

國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

適用於高速時脈產生之低功率全數位式頻率合成器

**A LOW POWER ADPLL-BASED FREQUENCY
SYNTHESIZER FOR HIGH SPEED CLOCK
GENERATION**

研 究 生：楊宗熙

指 導 教 授：黃 威 教 授

中 華 民 國 九 十 五 年 七 月

適用於高速時脈產生之低功率全數位式頻率合成器

**A LOW POWER ADPLL-BASED FREQUENCY
SYNTHESIZER FOR HIGH SPEED CLOCK
GENERATION**

研究生：楊宗熙

Student : Zong-Xi Yang

指導教授：黃 威 教授

Advisor : Prof. Wei Hwang



Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical Engineering and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Electronics Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月


適用於高速時脈產生之低功率全數位式頻率合成器

學生：楊宗熙

指導教授：黃 威 教授

國立交通大學電子工程學系電子研究所

摘 要



本論文提出一個新的數位控制頻率振盪器及一個新的相位頻率偵測器之架構以設計一個低功率的全數位式鎖相迴路。藉由使用一新型的數位控制延遲元件，此類數位控制頻率振盪器可具有其絕對單調的特性，且使的數位控制頻率振盪器的設計更為容易。此外，我們也提出了一個新的相位頻率偵測器，它可以在一個參考時脈週期內，完成頻率和相位的比較，並且能進一步調整振盪器的振盪頻率。

此全數位式頻率合成器是以 TSMC 0.13um 技術來做設計。它的輸出頻率範圍可從三百百萬赫茲到一千百萬赫茲，並且可以在十六個參考時脈週期內達到鎖定(最差的情況下)。輸出時脈訊號的鋒對鋒抖動值亦可維持在 120ps 之內。在供應電壓為 1.2 伏，操作頻率在 1 千百萬赫茲的情況下，此全數位式頻率合成器所消耗的總功率為 3.1 毫瓦。此外，參考現有的高速時脈應用之規格，此頻率合成器可作為高速數位訊號處理器的時脈產生器。

A LOW POWER ADPLL-BASED FREQUENCY SYNTHESIZER FOR HIGH SPEED CLOCK GENERATION

Student :Zong-Xi Yang

Advisor : Prof. Wei Hwang

Department of Electronics Engineering & Institute of Electronics

National Chiao-Tung University

ABSTRACT

This thesis proposes a new digital controlled oscillator (DCO) and a new phase frequency detector (PFD) architecture for the all digital phase-locked loop (ADPLL) with low power design. By using the new type digitally controlled delay element (DCDE), a digitally controlled oscillator (DCO) with characteristics of its monotonicity is presented, which makes the DCO design more straightforward. Besides, a new PFD architecture that can finish phase and frequency comparison and adjustment in one reference cycle is also presented.

The proposed ADPLL-based frequency synthesizer has been designed with TSMC 0.13um technology model. It can operate from 300 MHz to 1 GHz, and achieve frequency acquisition within sixteen reference clock cycles (worst case scenario). The peak-to-peak jitter of the output clock is less than 120 ps. Total power dissipation of the ADPLL-based frequency synthesizer is 3.1 mW at 1 GHz with a 1.2 V power supply. With the specification, it could be used for high speed clock generation in high speed DSPs applications.

Acknowledgements

I am grateful to have many people assisting in supporting this thesis. This thesis would not be accomplished without them.

First of all, I would like to thank my advisor, Prof. Wei Hwang, who has provided me a good research environment and the complete support, which does help me be devoted to the study of this research topic independently. With his edification and judicious advices, I was inspired in this knowledge field and also gained much significant experience.

Next, the fellows of my laboratory also help a lot on my thesis. They accompany me with delights and serious works, and it makes me own a colorful life in my years at NCTU.

Finally, I would also like to thank my family, for their concerns and understanding. With their encouragement and support, I could always stride forward without fear of disturbance in the rear. Thank you all.



Contents

Chapter 1 Introduction.....	1
1.1 Research Motivation	1
1.2 Thesis Organization	6
Chapter 2 An Overview of PLL	8
2.1 The Operating Principle of PLL.....	9
2.2 Linear PLL	11
2.3 Digital PLL	14
2.3.1 Phase Frequency Detector.....	15
2.3.2 Charge Pump/Loop Filter.....	18
2.3.3 Voltage Controlled Oscillator.....	20
2.3.4 Frequency Divider	23
2.4 All Digital PLL	25
2.5 An Example of The Conventional ADPLL Design.....	29
2.5.1 Architecture Overview	29
2.5.2 Digitally Controlled Oscillator	31
2.5.3 Frequency Acquisition.....	34
2.5.4 Phase Acquisition.....	37
2.5.5 Phase and Frequency Maintenance.....	39
Chapter 3 Digitally Controlled Oscillator.....	40
3.1 Basic Concepts of DCO	40
3.2 Different Approaches for DCO design.....	41
3.3 Digitally Controlled Delay Element	44
3.4 Digitally Controlled Oscillator Architecture.....	52
Chapter 4 A Low Power ADPLL Circuit Design	56
4.1 Architecture of The ADPLL.....	56
4.2 Circuit Design of The ADPLL	58
4.2.1 Phase/Frequency Detector	58
4.2.2 Control Unit	62
4.2.4 The Sub-Circuit Design	65
Chapter 5 The Implementation of Frequency Synthesizer	68
5.1 Frequency Synthesizer Architecture	68
5.1.1 Integer-N Architecture	69
5.1.2 Fractional-N Architecture	70

5.2 Frequency Dividers	73
5.2.1 Divide-by-Two Circuits	74
5.2.2 Dual-Modulus Dividers	75
5.3 The Proposed ADPLL-Based Frequency Synthesizer	78
5.4 Layout Implementation and Simulation Result	79
Chapter 6 Conclusion and Future Work	94
6.1 Conclusion	94
6.2 Future Work	95



List of Figures

Fig.1-1 Frequency multiplication.....	2
Fig.1-2 Frequency synthesizer	2
Fig.1-3 Skew between data and buffered clock	3
Fig.1-4 Use of a PLL to eliminate skew	4
Fig.1-5 (a) Retiming data with D flipflop driven by a low-noise clock	5
(b) use of a phase-locked clock recovery circuit to generate the clock	5
Fig.2-1 Block diagram of PLL.....	10
Fig.2-2 The transfer curve of VCO	10
Fig.2-3 The transfer curve of PD	11
Fig.2-4 Linear PLL model	12
Fig.2-5 Digital PLL Block	15
Fig.2-6 Phase Frequency Detector Block	16
Fig.2-7 (a) PFD response with input reference lagging feedback clock.....	16
(b) PFD response with freq. of input reference > that of feedback clock.....	16
Fig.2-8 PFD state diagram	17
Fig.2-9 Charge Pump and Loop Filter	18
Fig.2-10 three kinds of passive Loop Filter.....	19
Fig.2-11 The response of PFD and Charge Pump/Loop Filter	19
Fig.2-12 LC-Tank Voltage-Controlled Oscillator	22
Fig.2-13 five stage signal ended ring oscillator	23
Fig.2-14 Divide-by-two using a TSPC register	24
Fig.2-15 All Digital Phase-Locked Loop.....	26
Fig.2-16 The Conventional ADPLL Block Diagram	30
Fig.2-17 (a) The Constituent DCO cell, (b) The Digitally Controlled Oscillator....	33
Fig.2-18 The Frequency Comparator.....	36
Fig.2-19 Frequency Comparator Timing	37
Fig.2-20 (a) Phase Acquisition Mux, (b) Phase Acquisition Resolved.....	38
Fig.3-1 DCO composed of a DAC and a VCO.....	41
Fig.3-2 DCO composed of a high frequency oscillator and a divider	42
Fig.3-6 Basic structure of a delay element.....	46
Fig.3-7 Current-starved delay element	47
Fig.3-8 Another delay element.....	48
Fig.3-9 New DCDE architecture.....	50
Fig.3-10 The proposed DCO architecture.....	53

Fig.3-11 The Modified DCDE	53
Fig.3-12 DCO Frequency V.S. number of the input vector	54
Fig.4-1 The ADPLL Block Diagram.....	57
Fig.4-2 The Modified Binary Search	58
Fig.4-3 Phase/Frequency Detector	59
Fig.4-4 Timing diagram of the Phase/Frequency Detector	60
(a) unlock state, (b) lock state (for example: divider ratio=2)	60
Fig.4-5 DCO Enable Generator	61
Fig.4-6 (a) Block diagram of the Control Unit	63
(b) An example of the phase gain strategy	63
Fig.4-7 TSPC DFF	66
Fig.4-8 1-bit \overline{FA} (a) logic diagram, (b) transistor-level circuit	67
Fig.4-9 11-bit modified ripple adder/subtractor.....	67
Fig.5-1 Pulse swallow frequency divider.....	69
Fig.5-2 (a) Simple fractional-N synthesizer, (b) use of divider in the loop.....	72
Fig.5-3 Fractional-N synthesizer using a dual-modulus divider.....	72
Fig.5-4 Example of a fractional-N synthesizer	73
Fig.5-5 Divide-by-two circuit.....	74
Fig.5-6 Dynamic dividers using (a) inverters, (b) TSPC	75
Fig.5-7 (a) Divide-by-3 circuit (b) Divide-by-2/3 circuit	76
Fig.5-8 Divide-by-15/16 circuit	77
Fig.5-9 Adjustable DCO counter length	78
Fig.5-10 Layout of the ADPLL.....	80
Fig.5-11 Floor Plan of the ADPLL	81
Fig.5-12 Area Distribution of the ADPLL	81
Fig.5-13 Power-Consumption Distribution of the ADPLL.....	82
Fig.5-14 (a) The overall lock process of 300MHz target frequency.....	83
Fig.5-14 (b) The zoom in of lock process of 300MHz target frequency.....	83
Fig.5-14 (c) The locked state of 300MHz target frequency.....	84
Fig.5-15 (a) The overall lock process of 400MHz target frequency.....	84
Fig.5-15 (b) The zoom in of lock process of 400MHz target frequency.....	85
Fig.5-15 (c) The locked state of 400MHz target frequency.....	85
Fig.5-16 (a) The overall lock process of 500MHz target frequency.....	86
Fig.5-16 (b) The zoom in of lock process of 500MHz target frequency.....	86
Fig.5-16 (c) The locked state of 500MHz target frequency.....	87

Fig.5-17 (a) The overall lock process of 600MHz target frequency.....	87
Fig.5-17 (b) The zoom in of lock process of 600MHz target frequency.....	88
Fig.5-17 (c) The locked state of 600MHz target frequency.	88
Fig.5-18 (a) The overall lock process of 850MHz target frequency.....	89
Fig.5-18 (b) The zoom in of lock process of 850MHz target frequency.....	89
Fig.5-18 (c) The locked state of 850MHz target frequency.	90
Fig.5-19 (a) The overall lock process of 1GHz target frequency.....	90
Fig.5-19 (b) The zoom in of lock process of 1GHz target frequency.	91
Fig.5-19 (c) The locked state of 1GHz target frequency.....	91
Fig.5-20 Lock cycle VS. operating frequency	92
Fig.5-21 Jitter VS. operating frequency.....	92
Fig.5-22 Power dissipation VS. operating frequency	92
Fig.6-1 The ADPLL with AVS.....	96



List of Tables

Table 2-1 Comparison of different type oscillators	21
Table 2-2 Advantage and disadvantage of different type PLL.....	27
Table 2-3 Design issues of ADPLL.....	28
Table 3-1 Frequency Range of DCO with Different Environments	55
Table 5-1 Frequency of output clock V.S. input control signal.....	79
Table 5-2 Performance comparison of ADPLL	93
Table 6-1. Performance Summary of the Frequency Synthesizer.....	94



Chapter 1

Introduction

1.1 Research Motivation

The phase-locked loop (PLL) has been widely used in electronics, communication, and instrumentation today. Examples include memories, microprocessors, hard disk drive electronics, RF and wireless transceivers, and optical fiber receivers. In this section, we show some applications that demonstrate the versatility of phase locking. They are *Frequency Multiplication and Synthesis*, *Skew Reduction*, and *Jitter Reduction*, separately [1].

Frequency Multiplication A PLL can be modified such that it multiplies its input frequency by a factor of M . As shown in Fig.1-1, if the output frequency of a PLL is divided by M and applied to the phase detector, we have $f_{out}=M \cdot f_{in}$. From another point of view, since $f_D=f_{out} /M$ and f_D and f_{in} must be equal in the locked condition, the PLL multiplies f_{in} by M . The $\%M$ circuit is realized as a counter that produces one output pulse for every M input pulses.

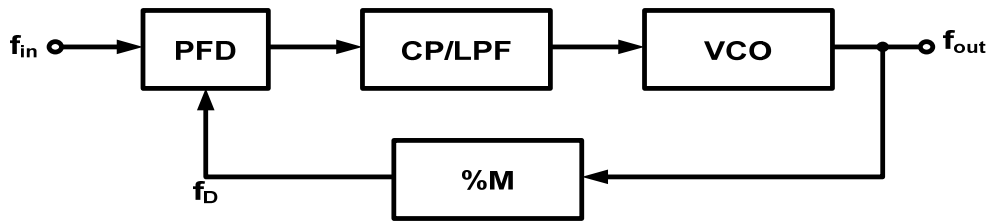


Fig.1-1 Frequency multiplication

The frequency-multiplying loop exhibits two interesting properties. First, the PLL provides a multiplication factor exactly equal to M . Second, the output frequency can be varied by changing the divide ratio M , an extremely useful property in synthesizing frequencies.

Frequency Synthesis Some systems require a periodic waveform whose frequency (a) must be very accurate (e.g., exhibit an error less than 10ppm), and (b) can be varied in very fine steps (e.g., in steps of 30 kHz from 900 MHz to 925 MHz). Commonly encountered in wireless transceivers, such requirements can be met through frequency multiplication by PLLs.

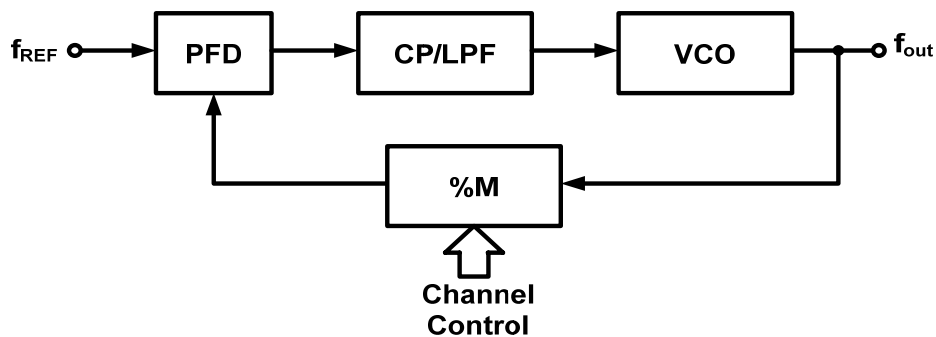


Fig.1-2 Frequency synthesizer

Fig.1-2 shows the architecture of a phase-locked frequency synthesizer. The channel control input is a digital word that varies the value of M . Since $f_{out}=M \cdot f_{REF}$, the relative accuracy of f_{out} is equal to that of f_{REF} . For this reason, f_{REF} is derived from a stable, low-noise crystal oscillator. Note that f_{out} varies in steps equal to f_{REF} if M changes by one each time.

CMOS frequency synthesizers achieving gigahertz output frequency have been reported. Issues such as noise, sidebands, settling speed, frequency range, and power dissipation continue to challenge synthesizer designers.

Skew Reduction The earliest usage of phase locking in digital systems was for skew reduction. Suppose a synchronous pair of data and clock lines enter a large digital chip as shown in Fig.1-3. Since the clock typically drives a large number of transistors and long interconnects, it is first applied to a large buffer. Thus, the clock distributed on the chip may suffer from substantial skew with respect to the data, an undesirable effect because it reduces the timing budget for on-chip operations.

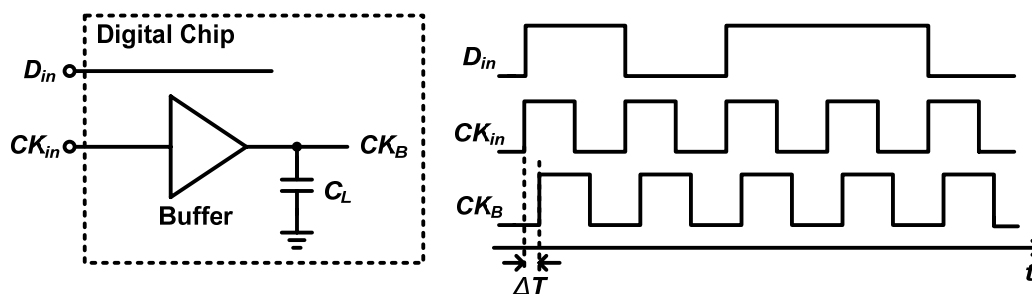


Fig.1-3 Skew between data and buffered clock

Now consider the circuit shown in Fig.1-4, where CK_{in} is applied to an on-chip PLL and the buffer is placed inside the loop. Since the PLL guarantees a nominally-zero phase difference between CK_{in} and CK_B , the skew is eliminated. From another point of view, the constant phase shift introduced by the buffer is divided by the infinite loop gain of the feedback system. Note that the VCO output, V_{VCO} , may not be aligned with CK_{in} , a nonetheless unimportant issue because V_{VCO} is not used.

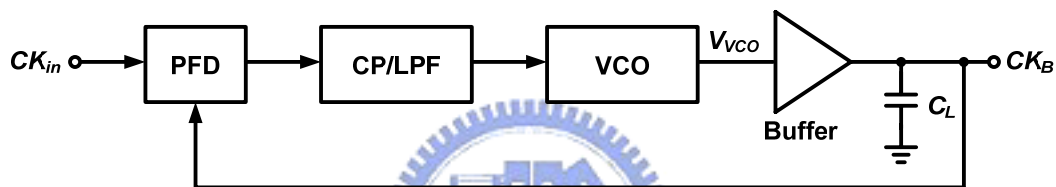
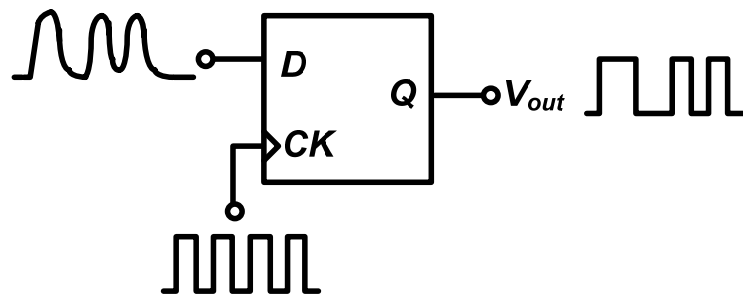


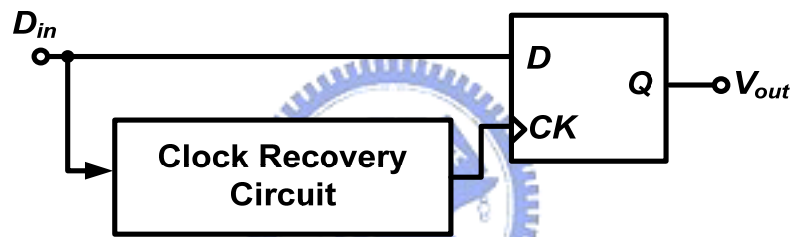
Fig.1-4 Use of a PLL to eliminate skew

Jitter Reduction Many applications must deal with jittery waveforms. Random binary signals experience jitter because of (a) crosstalk on the chip and in the package (b) package parasitics, (c) additive electronic noise of devices, etc. Such waveforms are typically “retimed” by a low-noise clock so as to reduce the jitter. Illustrated in Fig.1-5(a), the idea is to resample the midpoint of each bit by a D flipflop that is driven by the clock. However, in many applications, the clock may not be available independently. For example, an optical fiber carries only the random data stream, providing no separate clock waveform at the receive end. The circuit of Fig.1-5(a) is therefore modified as shown in Fig.1-5(b), where a “clock recovery circuit” (CRC) produces the clock from the data. Employing phase locking with a relatively narrow

loop bandwidth, the CRC minimizes the effect of the input jitter on the recovered clock.



(a)



(b)

Fig.1-5 (a) Retiming data with D flipflop driven by a low-noise clock

(b) use of a phase-locked clock recovery circuit to generate the clock

Phase locked loop (PLL) based clock generators for microprocessor are often required for on-chip clock generation and multiplication to produce several unrelated clocks with different frequency for other sub-systems. In traditional mixed mode circuit system, PLL is usually implemented in analog building block. Recently, the SoC (system-on-a-chip) architecture has become the underlying architecture for many embedded systems. That means conventional analog PLL integrated with digital circuits is inevitable. Integrating an analog circuit on a die with digital circuits,

however, has a large amount of generated digital noise. Besides, analog PLL is much more sensitive to process variation. It is too hard to use the same analog PLL design in different process [2], [3]. On the other hand, ADPLL are much easier to implement without targeting a specific technology. Their area would also scale down rapidly as the technology shrinks if only active components are used.

Since the implementation of analog component in a digital environment is not a simple task, the linear phase-locked loop (LPLL) and classical digital phase-locked loop (DPLL) which relay on analog component have been replaced by the all digital phase-locked loop (ADPLL) [4]-[7]. The ADPLL becomes more and more popular in recently year. In addition, the ADPLL has characteristics of fast frequency locking, full digitization, and good stability.



1.2 Thesis Organization

This thesis is organized as follows:

Chapter 2 gives an overview of PLL, including LPLL, DPLL and ADPLL, and also introduces an example of the ADPLL circuit design.

Chapter 3 first introduces the fundamentals of a digitally controlled oscillator (DCO). The different approaches including several digitally controlled delay elements (DCDE) for DCO design are also addressed. Then, we will focus on the new DCDE design and apply it as fine tune cell to build our DCO. Finally, the detailed description and simulation results of this circuit are given.

Chapter 4 presents design of the ADPLL with a modified phase/frequency detector

(PFD) architecture. Then, the detailed operation flow of the modified PFD will also be addressed. Besides, we also state each function circuit design in the control unit (CU).

Chapter 5 describes the fundamentals of frequency synthesizer and presents the ADPLL-based frequency synthesizer based on the adjustable counter length mechanism. Finally, we also show the implementation of layout, simulation result, and performance summary.

Chapter 6 gives the conclusion and future work.



Chapter 2

An Overview of PLL

In this chapter, we will review three kinds of phase locked-loop circuit, they are Linear PLL (LPLL), Digital PLL (DPLL), and All-Digital PLL (ADPLL), separately [4]. Then, we also introduce a design of the conventional ADPLL circuit, which is proposed by Motorola in 1995 [5].

The first PLL ICs appeared around 1965 and were purely analog devices. In the following years the PLL drifted slowly but steadily into digital territory. The very first digital PLL (DPLL), which appeared around 1970, was in effect a hybrid device. A few years later, the all-digital PLL (ADPLL) was invented. The ADPLL is exclusively built from digital function blocks and hence do not contain any passive components like resistor and capacitors. Different types of PLLs behave differently, roughly, the classifications of PLL circuit are defined as follows:

- (1) **LPLL (Linear PLL)**: Each block is analog.
- (2) **DPLL (Digital PLL)**: Phase Detector is digital and the others are analog.
- (3) **ADPLL (All Digital PLL)**: Each block is digital. The loop filter is from Up/Down counter. The Voltage Controlled Oscillator (VCO) is from Digital Controlled Oscillator (DCO).

2.1 The Operating Principle of PLL

A PLL is a circuit synchronizing an output signal (generate by an oscillator) with a reference or input signal in frequency as well as in phase. In the synchronized-often called locked-state the phase error between the oscillator's output signal and the reference signal is zero, or very small. If phase builds up, a control mechanism acts on the oscillator in such a way then the phase error is again reduced to a minimum. In such a control system the phase of the output signal is actually locked to the phase of the reference signal. This is why it is referred to as a phase-locked loop.

The operating principle of the PLL is explained by the example of the linear PLL [4]. In the Figure 2-1, the signals of interest within the PLL circuit are defined as follows:

$U_1(t)$: the reference signal

ω_1 : the angular frequency of the reference signal

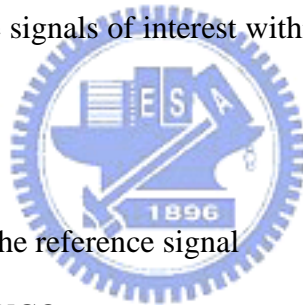
$U_2(t)$: the output signal of the VCO

ω_2 : the angular frequency of the output signal

$U_d(t)$: the output signal of the detector

$U_f(t)$: he output signal of the loop filter

θ_e : the phase error define as the phase difference between signals $U_1(t)$ and $U_2(t)$



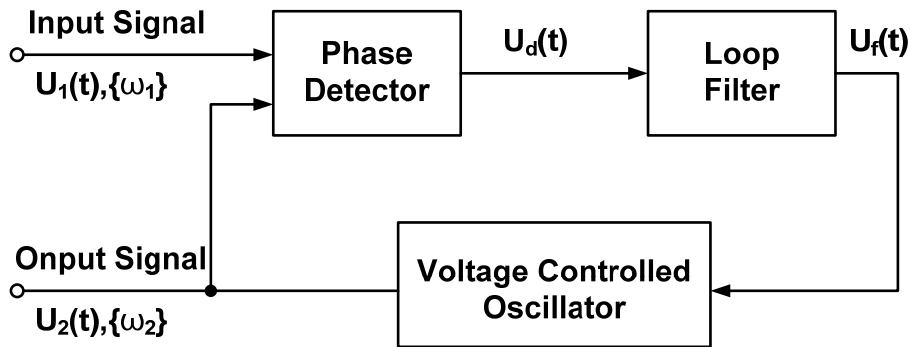


Fig.2-1 Block diagram of PLL

Now we look at the operations of the three functional blocks in the Figure 2-1.

VCO: VCO generate an angular frequency ω_2 , which is determined by the output signal U_f of the loop filter. The angular frequency ω_2 is given by equation (2.1), where ω_0 is the center frequency of the VCO and the K_0 is the VCO gain. Equation (2.1) is plotted graphically in the Figure 2-2.

$$\omega_2 = \omega_0 + K_0 \cdot U_f(t) \quad (2.1)$$

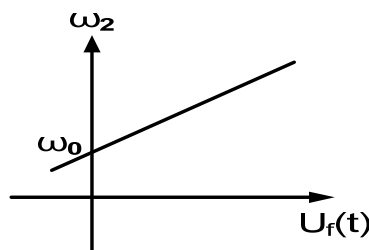


Fig.2-2 The transfer curve of VCO

Phase Detector: the Phase Detector compares the phase of the output signal of VCO with that of the reference signal and generate an output signal $U_d(t)$ which is approximately proportional to the phase error θ_e . Thus, we can write the equation as

equation (2.2), K_d is the gain of the phase detector. Equation (2.2) is plotted graphically in the Figure 2-3.

$$U_d(t) = K_d \cdot \theta_e \quad (2.2)$$

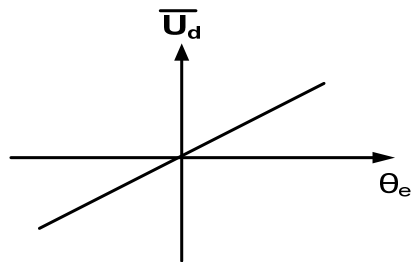


Fig.2-3 The transfer curve of PD

Loop Filter: Because the output signal of the PD consists ac component and it is undesired, so we need a loop filter to cancel the ac component.

Different types of PLLs have different building blocks. Following sections will introduce Linear PLL, Digital PLL and All Digital PLL The next section will discuss the design of Linear PLL.

2.2 Linear PLL

Although the PLL is a non-linear system, it can be described with a linear model if the loop is in lock [8]. When the loop is in lock the phase error signal generates by the phase detector settles on a constant value. In the locked state, the output signal has a fixed frequency as the input reference signal. A phase difference between the input

reference signal and output signal may exist depending on the type of PLL used. When the loop is in lock the phase difference remains constant. The Linear PLL is built from three purely analog function blocks. They are Phase Detector, Loop Filter and VCO. The three blocks are describe in the following:

Phase Detector: Phase Detector can be a four phase analog multiplier or analog signal mixer.

Loop Filter: Loop Filter is a passive or active RC filter, it filter high frequency signal and noise from phase detector and environment. The output of the filter is a DC value to send to VCO.

VCO: It is a ring oscillator which construct by inverters. The frequency is controlled by the dc value from Phase Detector.

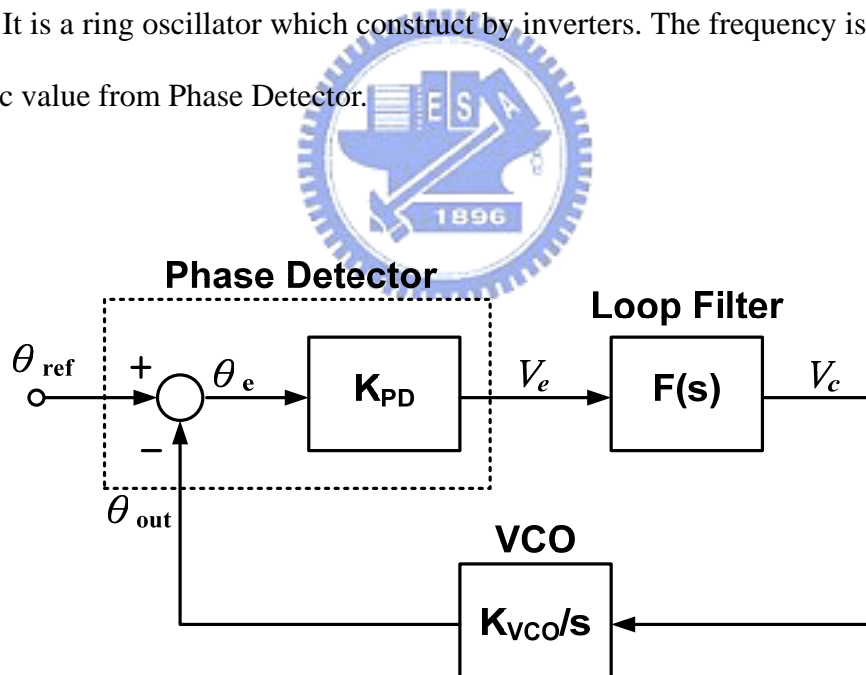


Fig.2-4 Linear PLL model

The building blocks of Figure 2-4 are taken as basis for the mathematical model of a Linear PLL in lock. From the model, we can derive the transfer function of the

Linear PLL:

$$H(s) = \frac{\theta_{out}(s)}{\theta_{ref}(s)} = \frac{K_{PD}K_{VCO}F(s)}{s + K_{PD}K_{VCO}F(s)} \quad (2.3)$$

The phase error transfer function is equal to the following:

$$H(s) = \frac{\theta_e(s)}{\theta_{ref}(s)} = \frac{s}{s + K_{PD}K_{VCO}F(s)} \quad (2.4)$$

The VCO control voltage transfer function is equal to the following:

$$H(s) = \frac{V_C(s)}{V_{ref}(s)} = \frac{sK_{PD}F(s)}{s + K_{PD}K_{VCO}F(s)} \quad (2.5)$$

The following observation is made from the transfer function give in equation (2.3), (2.4) and (2.5). At first, we discuss the Linear PLL transfer function, give in equation (2.3), it has a low-pass characteristic. This means that for slow (low frequency) variations in the reference phase, the loop will basically track the input signal and produce an output phase.

The phase error transfer function, give in equation (2.4), has a high pass characteristic. This implies that for slow variations in the reference phase, the phase error will be small. However, fast variations in the reference phase will not be filtered and show up as a phase error.

The VCO control voltage transfer function, give in equation (2.5), also has a high pass characteristic. However, depending on the parameter of the loop filter, it can take on a more band-pass shape.

The linear model in Figure 2-4 enables us to analyze the tracking performance of the Linear PLL, i.e., the system maintains phase tracking when excited by phase steps, frequency steps, or other excitation signals. So we can analyze the characteristic and the responses of the Linear PLL in S-domain, and then, calculate all parameter to design a Linear PLL to satisfy the specification.

2.3 Digital PLL

In this section, we will describe the operating principle and circuit design of Digital PLL [4]. Figure 2-5 shows the Digital PLL which consists digital Phase Frequency Detector, analog Charge Pump, analog Loop Filter, analog Voltage Controlled Oscillator and Frequency Divider.

The Phase Frequency Detector can detect the phase and frequency error between the input reference signal and feedback clock signal. The output of the PFD is up signal or down signal. The up signal and down signal control the Charge Pump to charge or discharge. Loop Filter can filter the high frequency signal. Loop Filter outputs a low frequency signal to control the VCO. By including a Frequency Divider in the feedback path, the VCO output clock runs N times faster than the feedback clock. The next sections will describe the circuit and behavior of the PFD, CP, LF, FD, and VCO.

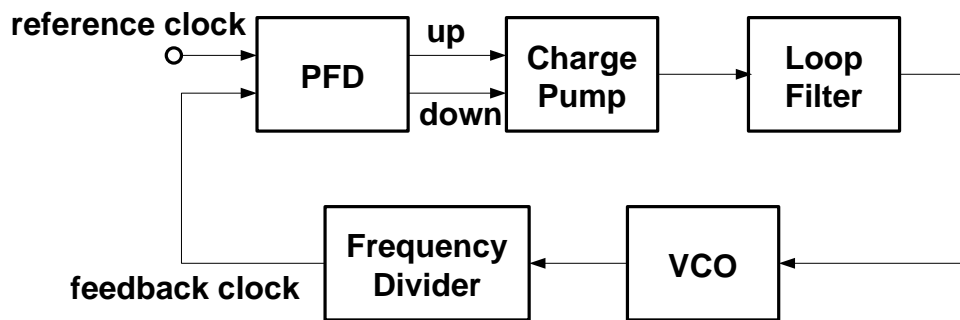


Fig.2-5 Digital PLL Block

2.3.1 Phase Frequency Detector

This section will describe the operation and implementation of the PFD circuit. Figure 2-6 shows an example of the PFD circuit and Figure 2-7 shows the waveforms in some conditions. Unlike multipliers and XOR gate, sequential PFD generates two outputs that are not complementary. Illustrated in Figure 2-6, the operation of a typical PFD is as follows.

When the feedback clock is high and the input reference is low, then the PFD produces positive pulses at down signal, while up signal remains at zero. Conversely, if input reference is high and feedback clock is low then positive pulses appear at up signal while down signal is zero. It should be note that, in principle, up and down are never high together in the simulation. The average value of up – down is an indication of the frequency or phase difference between input reference and feedback clock.

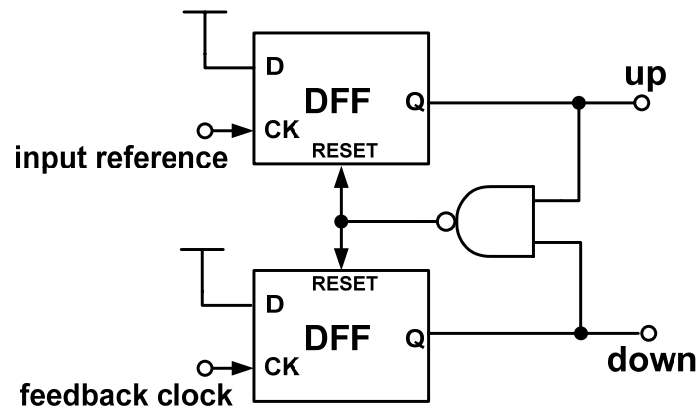
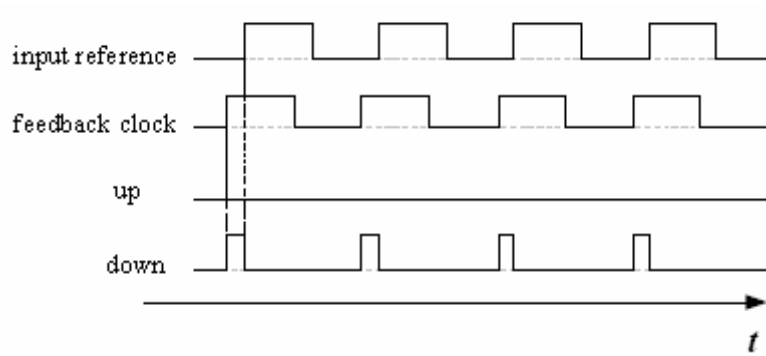
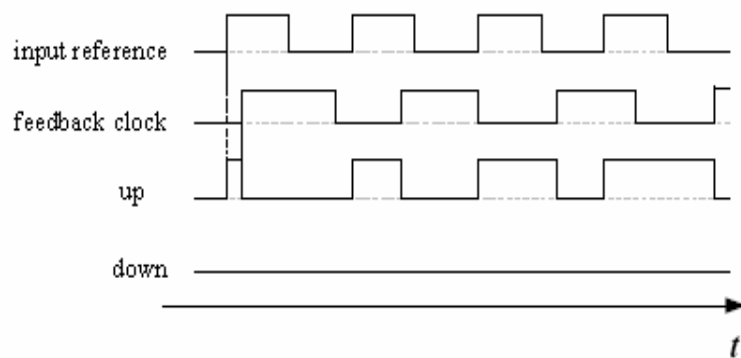


Fig.2-6 Phase Frequency Detector Block



(a)



(b)

Fig.2-7 (a) PFD response with input reference lagging feedback clock

(b) PFD response with freq. of input reference $>$ that of feedback clock

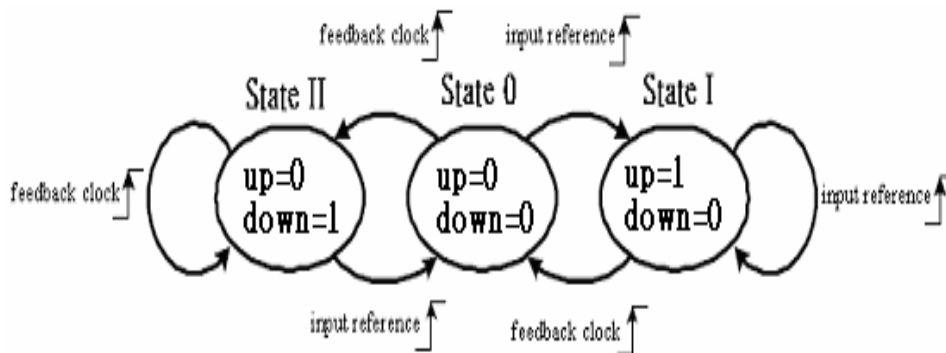


Fig.2-8 PFD state diagram

In the Figure 2-8, it shows the PFD circuit behavior. It has three state diagrams: up=1,down =0(state I) ; up=0,down=0(state 0) ; up=0,down=1(state II). Because the PFD is buildup from two edge-triggered sequential circuits, we can avoid dependence of the output upon the duty cycle of the inputs.

If the PFD is in the state 0, up=down=0, then a transition on A take it to state I, where up=1, down=0. With state I is reached, any more rising edges at input A won't cause state change at all. The circuit will remain in this state until a transition occurs on B, upon which the PFD returns to state 0. The switching sequence between state 0 and state II is similar.

The PFD can nominally detect a full range of phase difference, i.e. $+2\pi$, -2π . A phase difference larger than 2π is truncated with respect to integer of 2π . The output of the PFD can drive a three-state charge pump. The charge pump and loop filter will be discussed followed.

2.3.2 Charge Pump/Loop Filter

In a PLL system, the charge pump transfers the digital signal of up and down from the PFD to an analog signal. Figure 2-9 shows a simple model of the charge pump circuit. It consists of both matched current sources, each with a fixed value. When the up signal is high, the switch connects to A and V_c is charged by the up current source I_{up} . Similarly, when the down signal is high, the switch connects to B and V_c is charged by the lower current source I_{down} . If both up signal and down signal are low, then the switch maintains at original node and V_c holds the original voltage.

Most of the PLL's specifications are determined by the loop filter. The loop filter can be either passive or active. In general, a passive filter is simple to design and has better noise performance. The passive filter was shown in Figure 2-10, which may be first-order, second-order, or other high order structures.

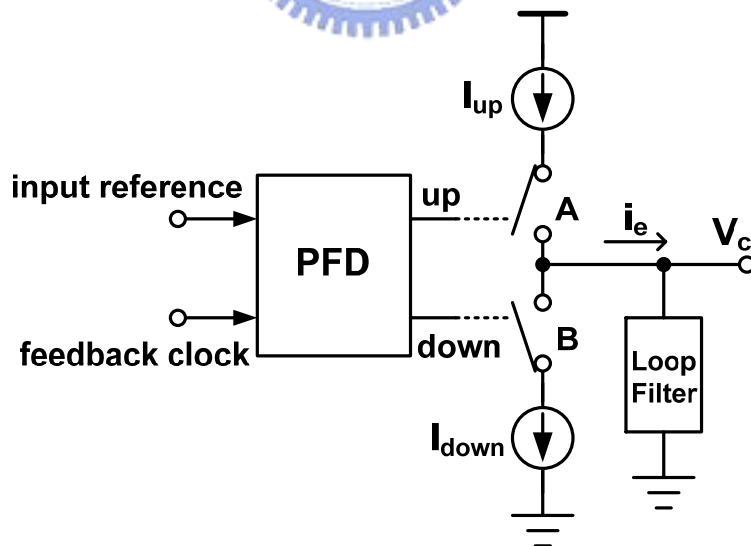


Fig.2-9 Charge Pump and Loop Filter

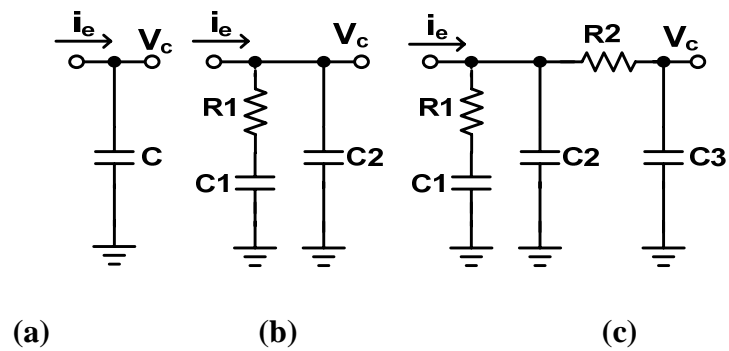


Fig.2-10 three kinds of passive Loop Filter

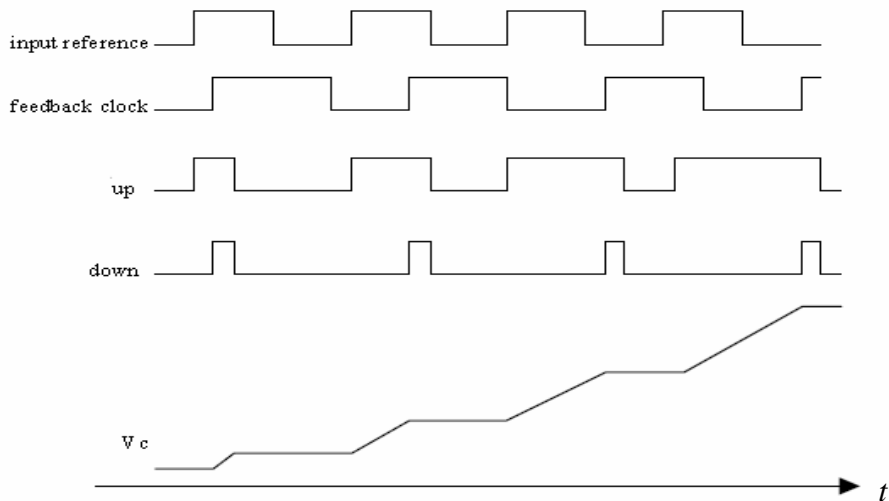


Fig.2-11 The response of PFD and Charge Pump/Loop Filter

As show in Figure 2-11, charge pump circuit convert the logic state of the PFD (Up and Down) into an analog counterpart for controlling the VCO. The charge pump output and the input of a VCO must have the low leakage tendency. So a passive loop filter shapes the output of the charge pump circuit to suppress the un-wanted message. The time domain response can be shown in Figure 2-11.

As discussed in the previous section, if the input reference signal leads the feedback signal, the pulse appear at up signal, then positive charge accumulates on capacitor steadily. Conversely, if the input reference signal lags the feedback signal, the charge is removed from capacitor on every phase comparison. In the third state, when input reference and feedback signal are equal, up and down keep low. Both switches are off, and the output signal V_c remains constant.

The above discussions of the Figure 2-11 only use a capacitor as the loop filter. But this kind of filter makes the PLL unstable. We can use the loop filter which was shown in Figure 2-10(b), Figure 2-10(c) to avoid instability.



2.3.3 Voltage Controlled Oscillator

In this section, we will describe the voltage-controlled oscillator which is the critical circuit in the PLL. The input voltage of the VCO generated from the loop filter and the output frequency signal of VCO is controlled by the input voltage. In some oscillators, the frequency of the oscillator is controlled by a current rather than a voltage. They are referred as current-controlled oscillators (CCO) and play the same role as those of VCOs in PLLs. The VCO and CCO are similar. Of course, there are various types of VCO than can be used in PLLs.

The Table 2-1 show three various types of VCO. Basically, the VCO has to fulfill some constraints is the phase noise in the frequency domain or the timing jitter in the time domain. Other important factors are the bandwidth of the VCO, linearity of the controlled voltage, output voltage swing and the power consumption [9][10][11].

Table 2-1 Comparison of different type oscillators

<i>Type</i>	<i>Advantage</i>	<i>Disadvantage</i>
<i>Voltage controlled crystal oscillators</i>	Phase accuracy, good noise performance	Cannot be integrated and cost is high and low frequency
<i>Ring oscillator VCOs</i>	Suitable for integration and have wide control range	Poor jitter performance
<i>LC-tuned oscillators</i>	High frequency and good noise performance	The inductor is difficult to integrate and cost high

Some of the most important considerations of VCO are: [12]

(1) Phase Stability:

The frequency spectrum of a VCO output should look like an ideal impulse, i.e., the phase noise of a VCO must be as low as possible.

(2) Electrical Tuning Range:

The tunable frequency range of a VCO must be able to cover the entire required frequency range of the interested application.

(3) Tuning Linearity:

An ideal VCO has a constant gain at the entire frequency range. Also, a constant VCO gain can simplify the design procedure of a VCO.

(4) Power Supply Sensitivity:

Since there are many digital circuits in a modern transceiver circuit, the switching activities of digital circuits will somewhat influence the power supply of the whole system. The switching noise induced by digital circuits will also couple to the power

supply of the VCO and influence its output waveform. Therefore, in VCO the dependency of the oscillating frequency on the power supply must be as low as possible.

(5) Frequency pushing:

The dependency of the center frequency on the power supply voltage.

(6) Frequency pulling:

The dependency of the center frequency on the output load impedance.

(7) Low cost, Phase noise, DC consumption current, Harmonic/spurious

In the next, we will show an LC-Tank VCO and a ring oscillator in the Figure 2-12 and Figure 2-13 [11]. In the LC-Tank VCO, the oscillation conditions are already shown in [11], its operation frequency is $\omega_0 = \frac{1}{\sqrt{LC}}$. Where R_p is the parasitic resistance in parallel to the LC-tank, and R_L and R_C are the parasitic resistances of L and C, respectively.

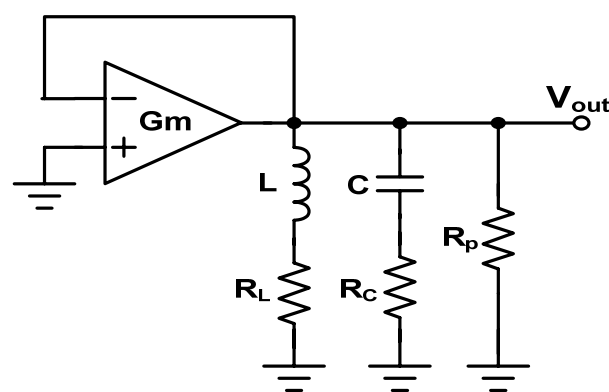


Fig.2-12 LC-Tank Voltage-Controlled Oscillator

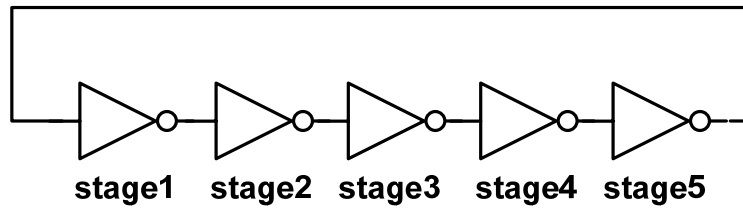


Fig.2-13 five stage signal ended ring oscillator

The second type oscillator is ring oscillator as shown in Figure 2-13 has been widely used in PLL for application of clock recovery and clock generation before, it can be smoothly integrated in a standard CMOS process without taking extra processing steps because it does not require any passive resonant element.

When the ring oscillator is employed as a voltage controlled oscillator, the desired wide operating frequency range can be easily obtained. Different output frequency is achieved by adjusting the timing delay of each stage in the ring oscillator.

The other category of oscillator is to eliminate that the real part of the loop's impedance so that the poles are pure imaginary. The LC-tank VCO is a typical resonator oscillator that bases on the idea and is called resonator oscillator. The VCO is the most challenging part of the PLL and we have to design carefully.

2.3.4 Frequency Divider

In some application, we need a high frequency clock generator and the crystal-oscillator is not satisfied, because the frequency of the crystal-oscillator is too

small. Therefore, the multiple-frequency-technology that utilizes PLL is presented. For example, if the divider module is four, then the output frequency of VCO is a four times of the input reference signal's frequency. The Figure 2-14 shows an example of divider, which uses a true single-phase clocking (TSPC) register. If we need higher division, it can be achieved by simple cascading divide-by-2 stages. The next is the advantages and the disadvantages of the divider:

Advantages:

- (1) Reasonably fast
- (2) No static power consumption
- (3) Compact size
- (4) Differential clock not require

Disadvantages:

- (1) Slowed down by stacked PMOS, signals goes through three gates per cycle
- (2) Requires full swing input clock signal

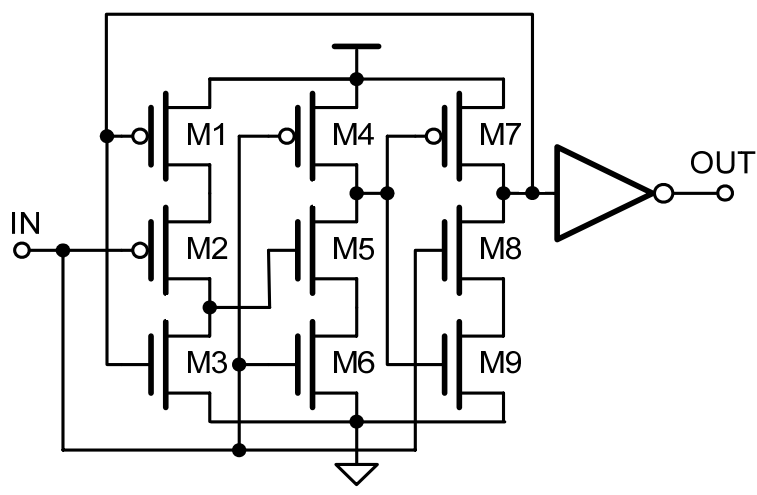


Fig.2-14 Divide-by-two using a TSPC register

2.4 All Digital PLL

In this section we will describe All Digital PLL, which has characteristics of fast frequency locking, full digitization and good stability. Because of the availability of low-cost ADPLL ICs, this type of PLL can replace the classical DPLL in many applications today. The ADPLL is made as a digital building block, it does not contain any passive component, such as resistors and capacitors.

The ADPLL consists of a digital phase frequency detector (PFD), a control unit, a frequency divider, and digital control oscillator (DCO) as shown in Figure 2-15. All signals in the ADPLL are digital signals. The PFD detects the frequency difference and the phase difference between the input reference signal and the feedback signal. The control unit receives the signal, produced by the PFD, and produces a set of digitally controlled signals to control the DCO.

By including a divide-by-N divider in the feedback path, the DCO output frequency runs N times faster than input reference signal. The divide-by-N divider is an optional component in the ADPLL. The functional blocks of the ADPLL imitate the function of the corresponding analog blocks. Because the ADPLL consists of digital circuits entirely, there are many different of design methods to achieve the functions of them.

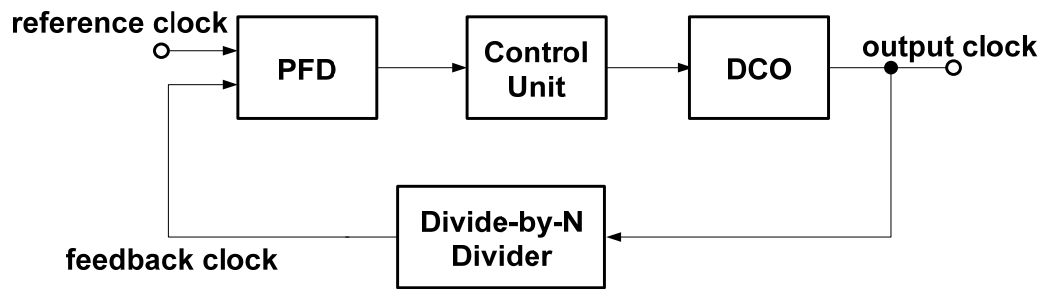


Fig.2-15 All Digital Phase-Locked Loop

The ADPLL system is a discrete-time system, hence analyzing the ADPLL in s-domain is not suitable. Although it is possible to take an entire PLL-description and then transform it from s-domain into z-domain, this is unnecessary difficult. Instead, one transforms each component into z-domain and then proceeds with the analysis in z-domain. The ADPLL is best described in z-domain.

In the Linear PLL, Digital PLL and All Digital PLL, they have many advantages and disadvantages respectively. We summarize them in Table 2-2. As shown in the Table 2-2, we can know that they use different design methodology, because the ADPLL is a digital circuit design so it can be designed by standard cell library. Hence the ADPLL just need a short design cycle than the analog architecture.

The ADPLL also has higher noise immunity than LPLL and DPLL. The VCO of the LPLL or the DPLL produces a continuous frequency band but the DCO of the ADPLL produces a discrete frequency band, so the VCO has higher resolution than DCO. In general, the ADPLL have a less power consumption than analog architecture because of using digital circuits. Besides, since the loop filter of LPLL or DPLL has one or more large capacitors, whose area can not be efficiently reduced as the process technology improving. In addition, the ADPLL can shorten lock time by dealing with

digital signal.

The design of PLL is a trade-off between jitter performance, frequency resolution, phase resolution, lock-in time, area cost, power consumption, circuit complexity and design cycle. It is hard to design one PLL suitable for all applications. For fast-locking frequency synthesizer applications, such as a frequency hopping multiple access system, the lock-in time is the most critical design issue. And for portable or mobile applications, lock-in time is also very important since the PLL must support fast entry and exit from power management techniques.

Table 2-2 Advantage and disadvantage of different type PLL

	<i>LPLL</i>	<i>DPLL</i>	<i>ADPLL</i>
<i>Design methodology</i>	Analog	Mixed mode	Digital
<i>Design cycle</i>	Slow	Slow	Fast
<i>Noise rejection</i>	Poor	Poor	Good
<i>Output frequency</i>	High	High	Low
<i>Oscillator resolution</i>	High	High	Low
<i>Lock cycle</i>	Slow	Slow	Quick
<i>Power consumption</i>	Large	Large	Small
<i>Area</i>	Large	Large	Design dependent

In traditional analog PLL designs, fast acquisition requires tuning of the VCO free-running frequency near the desired the frequency in advance or to increase loop bandwidth. But increasing the loop bandwidth degrades jitter performance, and the extra VCO tuning range is not easy to be achieved since there always has process variations, voltage variations, and temperate variations (PVT variations). Some

critical issues in ADPLL design are listed in the following table.

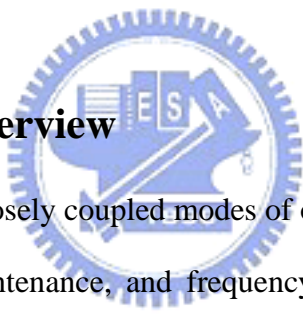
Table 2-3 Design issues of ADPLL

<p><i>Digital Controlled Oscillator</i></p>	<ul style="list-style-type: none"> (1) The output clock of DCO is discrete, so the resolution of a DCO should be sufficiently high to maintain acceptable jitter. (2) For searching target frequency and phase easily and efficiently, a DCO had better approach a monotonic response to the control word. (3) A DCO had better have high noise immunity, so the output clock will not induce large jitter.
<p><i>Phase Frequency Detector</i></p>	<ul style="list-style-type: none"> (1) The resolution of a PFD had better be as high as possible. In this way, the PFD can detect tiny phase difference to promote accuracy and to decrease jitter. (2) The PFD had better have two properties simultaneously. One is to judge the modulating direction. The other is to judge the modulating magnitude.
<p><i>Control Unit</i></p>	<ul style="list-style-type: none"> (1) Control Unit receives the signal, produced by the PFD, and produces signal to the DCO. It works as a loop filter. It both decides the speed of the lock process and suppresses the high frequency noise to reduce jitter. All responses of an ADPLL are almost decided by this control unit.

2.5 An Example of The Conventional ADPLL Design

In this section, we will introduce a design of the conventional ADPLL, which is proposed by Motorola in 1995 [5]. It has a 50-cycle phase lock, has a gain mechanism independent of process, voltage, and temperature, and is immune to input jitter. A DCO forms the core of the ADPLL and operates from 50 to 500 MHz, running at 4X the reference clock frequency. The DCO has 16b of binary weighted control and achieves LSB resolution under 500fs.

2.5.1 Architecture Overview



The ADPLL uses four loosely coupled modes of operation: frequency acquisition, phase acquisition, phase maintenance, and frequency maintenance. The phase-lock process is separated into frequency acquisition and phase acquisition, which significantly reduces the phase-lock time penalty.

Fig. 2-16 depicts a block diagram of the ADPLL. The DCO control register holds the 16 b, binary weighted DCO control word, which dictates the frequency of the DCO. Arithmetically incrementing or decrementing the DCO control word modulates DCO frequency and phase. The adder and subtractor provide the updates to the DCO control register. Also, the anchor circuit, which contains a register and an adder, updates the DCO control register in frequency maintenance mode. The frequency-gain register and phase-gain register provide operands to the adder and subtractor via the add mux and subtract mux. In addition, the phase-gain register provides data to the

anchor circuit. The control block marshals these sub-blocks to implement the different ADPLL modes of operation.

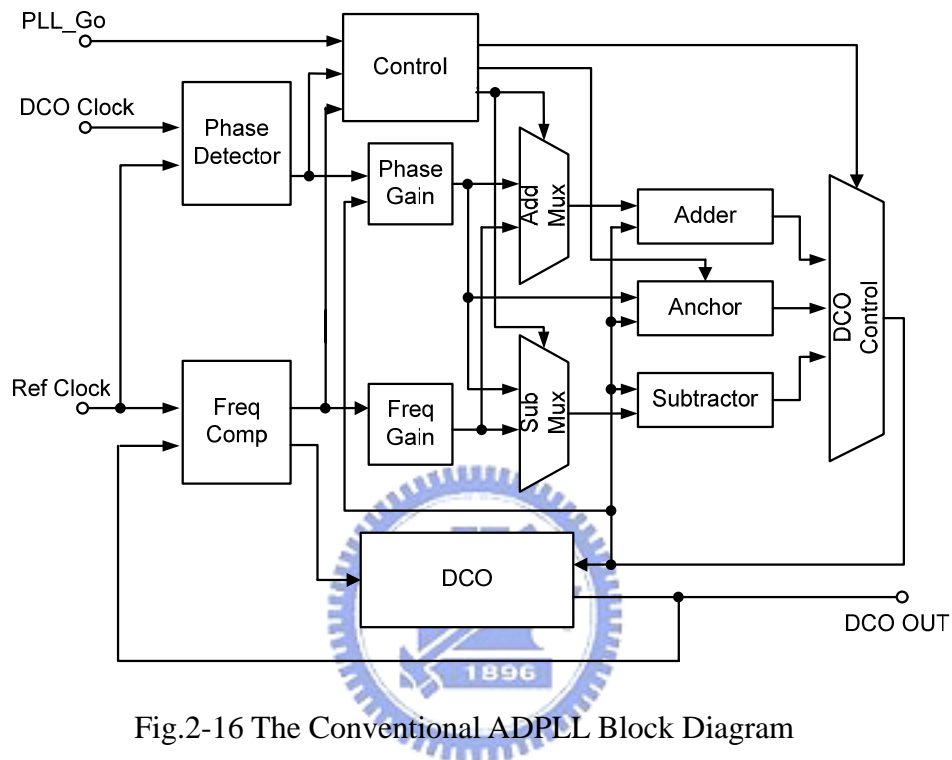


Fig.2-16 The Conventional ADPLL Block Diagram

Phase lock begins with frequency acquisition. In this mode an algorithm sweeps the DCO frequency range (divided by 4) to match that of the reference clock. The algorithm makes incremental changes to the DCO control word based on the output of the frequency comparator. The value held in the frequency-gain register determines the magnitude of the incremental changes. At the end of frequency acquisition, the ADPLL transfers the DCO control word defining the correct (baseline) frequency to the anchor register.

When frequency acquisition is complete, the ADPLL enters phase acquisition mode. During phase acquisition, the ADPLL increments or decrements the DCO

control word until the phase detector senses a change in the phase polarity of the reference clock relative to the internal clock. The value held by the phase-gain register dictates the magnitude of the changes to the DCO control word in phase acquisition mode, as well as in phase and frequency maintenance modes. Phase acquisition is finished when a change in phase polarity occurs. To complete the phase-lock process, the anchor register transfers its contents to the DCO control register, restoring the DCO control word value representing the baseline frequency.

After the phase-lock process, frequency maintenance and phase maintenance will operate concurrently. In frequency maintenance mode, an algorithm increments or decrements the content of the anchor register, changing the baseline frequency value. In phase maintenance mode, the ADPLL increments or decrements the DCO control word every reference cycle, based on the output of the phase detector, unless the polarity of phase error changes from that of the prior cycle. When phase polarity changes, the anchor register transfers its contents to the DCO control register to restore the baseline frequency. The ADPLL varies the magnitude of changes to the DCO control word, which changes the ADPLL response from overdamped to underdamped and vice versa as necessary.

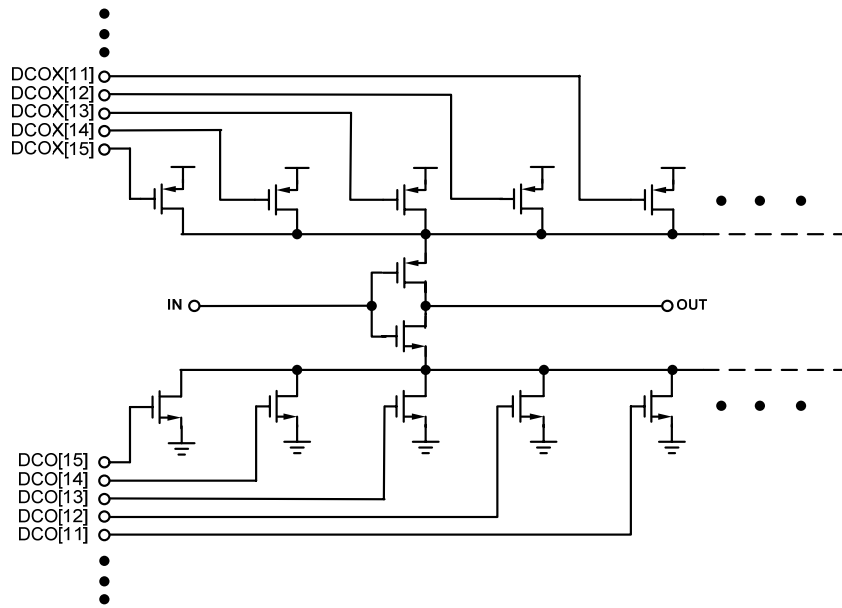
2.5.2 Digitally Controlled Oscillator

At the heart of the ADPLL is a digitally-controlled oscillator (DCO). The ADPLL controls the DCO frequency through the DCO control word, the output of a register consisting of sixteen binary weighted control signals. The requisite odd

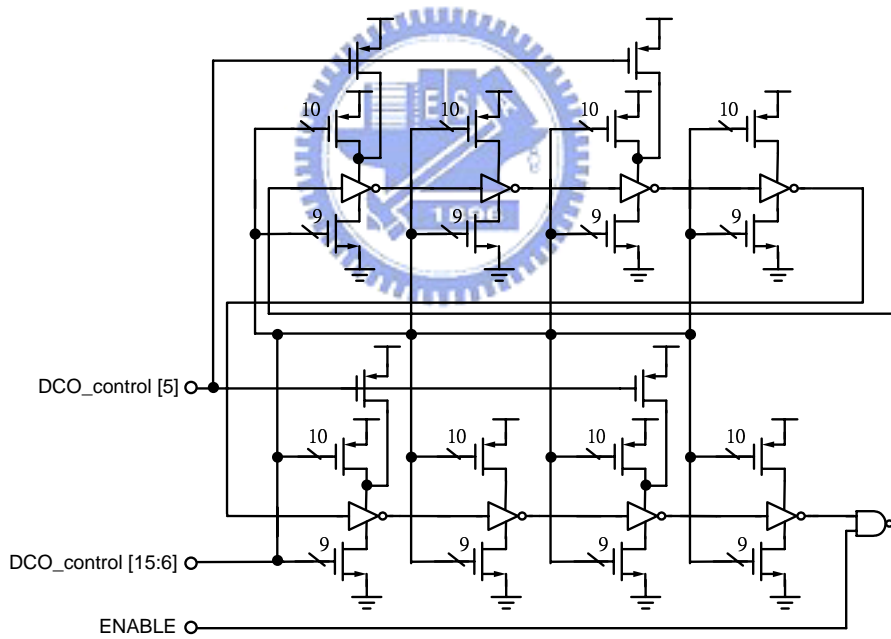
number of inverting stages in the DCO is obtained by using one enabling NAND gate and eight controllable cells. A mux selects between using four of the controllable DCO cells rather than eight to increase the range of the DCO. This selection occurs based on the first frequency comparison at the beginning of phase lock. Figure 2-17(a) illustrates the constituent DCO cell. The sizing ratio of the control devices is 2x, achieving binary weighted control. The most significant control-word bit (15) corresponds to the largest control device.

A key design criterion in DCO circuit design is to provide sufficient control word resolution to maintain acceptable jitter. With a minimum device width of 1.2 μm and a maximum of $256 \times 1.2 \mu\text{m}$, the DCO can achieve 9 b of control in each of 8 cells. Besides, using a stand-alone, minimum width PMOS device (and eliminating the corresponding pull-down device) adds a further bit of resolution in each cell. Taking the same PMOS device and using it in only 4 of 8 DCO cells adds yet another bit of resolution.

Fig. 2-17(b) depicts this concept, showing the DCO where control bit 5 affects four DCO cells and higher order bits affect eight DCO cells. Similarly, two more bits of resolution result from using the same PMOS device in only 2 of 8 and 1 of 8 DCO cells respectively. Finally, using the same width PMOS device but increasing the channel length-each used in only 1 of 8 cells-results in three more bits of resolution. Empirically tuning these devices compensates for the increase in channel-to-gate capacitance associated with longer lengths. Overall, these techniques yield 16 b of resolution.



(a)



(b)

Fig.2-17 (a) The Constituent DCO cell, (b) The Digitally Controlled Oscillator

2.5.3 Frequency Acquisition

The goal of frequency acquisition mode is to lock the DCO frequency (div4) to that of the reference clock frequency, ignoring phase alignment. A modified binary-search algorithm sweeps the DCO frequency range in progressively smaller increments. The implementation of this algorithm centers around a frequency-gain register and a frequency comparator.

The frequency-acquisition algorithm begins with initializing the DCO control register and the frequency gain register. A frequency comparator then performs the first frequency comparison of the DCO output frequency relative to the reference clock frequency. By default, a mux selects the 8-cell DCO configuration. However, if the first comparison indicates the DCO frequency is slow, then the mux selects the 4-cell DCO. The comparator then performs the next frequency comparison. Based on the results of the comparison, the adder or subtracter increments or decrements, respectively, the DCO control register by the value in the frequency gain register.

If the comparator output changes from fast to slow (or vice versa) over two consecutive frequency comparisons, then a change in search direction has occurred. The algorithm reduces frequency gain on every change in search direction. The algorithm proceeds with successive frequency comparisons until the frequency gain falls below the value of the DCO control word shifted right by ten places. Finally, the algorithm loads the anchor register with the DCO control word value matching the baseline frequency.

In implementing the frequency comparison technique of the algorithm, a comparator accepts as inputs the reference clock and the DCO output. It generates two mutually exclusive output signals, “slow” and “fast,” and also an enable signal for the

DCO. The comparator uses the reference clock edge to assert the DCO enable, forcing initial phase alignment of the DCO output edge to the reference clock edge. The comparator takes one-half reference cycle for synchronization before asserting “slow” or “fast.” Resetting the comparator to disable the DCO requires the remainder of the reference cycle. Hence, a complete frequency-comparison iteration takes two reference cycles.

For example, Fig. 2-18 shows a block diagram of the frequency comparator and Fig. 2-19 shows a timing diagram of two frequency comparison iterations. The first rising edge of the reference clock enables the DCO at **A**, and the second rising edge captures the output of a 4-b counter at **B**, the input to the synchronizer. If this signal at **B** is asserted when the second rising reference edge arrives at the synchronizer, then the DCO is fast. Conversely, if the signal at **B** is not asserted when the second rising reference edge arrives at the synchronizer, then the DCO is slow.

After synchronization, the comparator outputs a “fast” or “slow” at **C**. Concurrently, the comparator disables the DCO at **D**, and the ADPLL loads the new control word. The circuit at **E** matches (via circuit replication) the delay inherent in enabling the DCO and the delay inherent in the DCO pulse counter. If the frequency of the DCO (div4) identically matches the frequency of the reference clock, then both the signal and the synchronizer clock arrive simultaneously at the synchronizer (**B**) with the same rise times.

In implementing the gain strategy of the algorithm, an adder and subtracter receive both the value of a frequency-gain register (via the add and subtract muxes) and the DCO control word. Shifting the frequency-gain register once to the right decreases the gain by a factor of two. In this implementation, the add mux receives

only the odd bits of the frequency-gain register (the add gain), and the subtract mux receives only the even bits of the frequency-gain register (the subtract gain). In addition, setting the two most significant bits of the frequency-gain register initializes the add gain to 4000_{16} , and the subtract gain to 2000_{16} .

Now that the DCO frequency is locked to that of the reference clock, the DCO enable remains asserted so that the DCO is in the free-running state, ready for phase-acquisition.

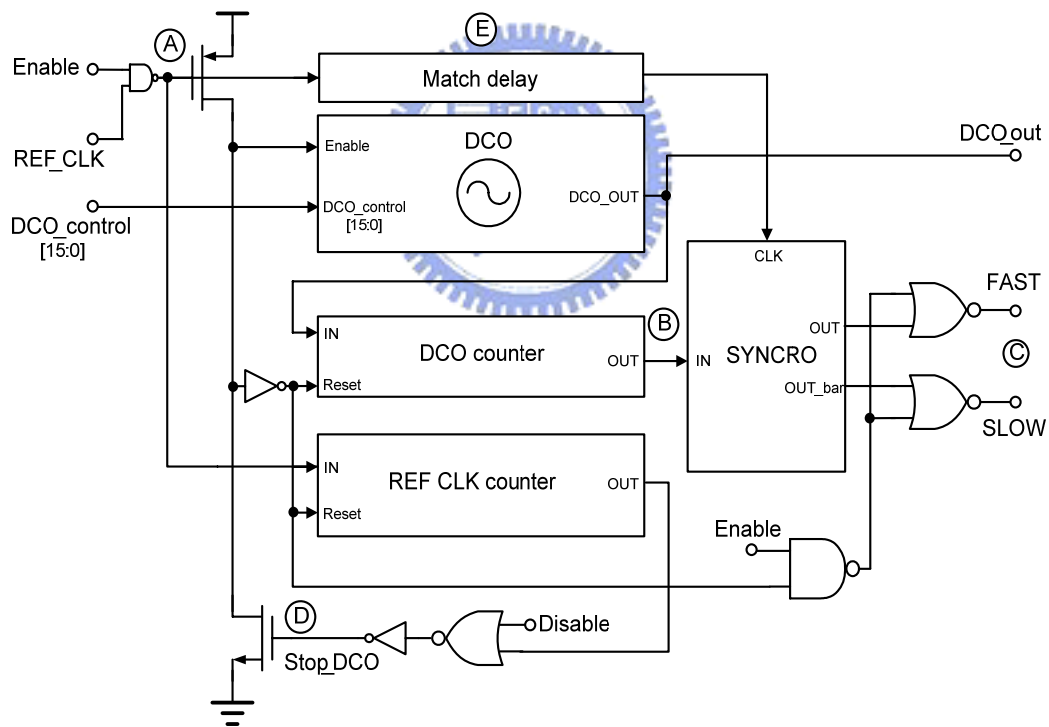


Fig.2-18 The Frequency Comparator

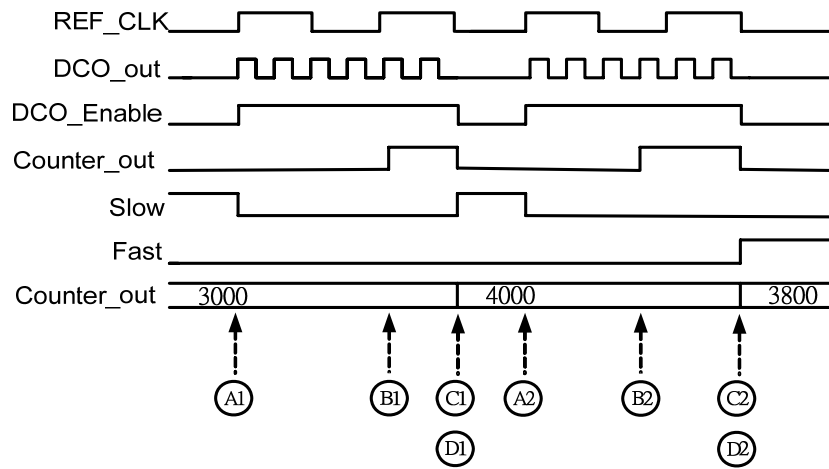


Fig.2-19 Frequency Comparator Timing

2.5.4 Phase Acquisition



The goal of this mode is to align the DCO clock edge to the reference clock edge. Because, in practice, there are several stages of logic separating the DCO clock from the DCO output. The implementation of this algorithm centers around a phase detector and a phase-gain register.

The phase-acquisition algorithm begins by selecting the phase gain and deselecting the frequency gain via the add and subtract muxes. The phase detector asserts a digital signal, either "ahead" or "behind," based on the relation of the DCO clock edge to the reference clock edge. This event increments or decrements the DCO control word by the value in the phase-gain register, thereby modulating the DCO phase relation.

When the output of the phase detector switches from "ahead" to "behind" or vice

versa on successive cycles, phase acquisition is complete, and the ADPLL transfers the anchor register contents to the DCO control register, restoring the baseline frequency and completing phase lock.

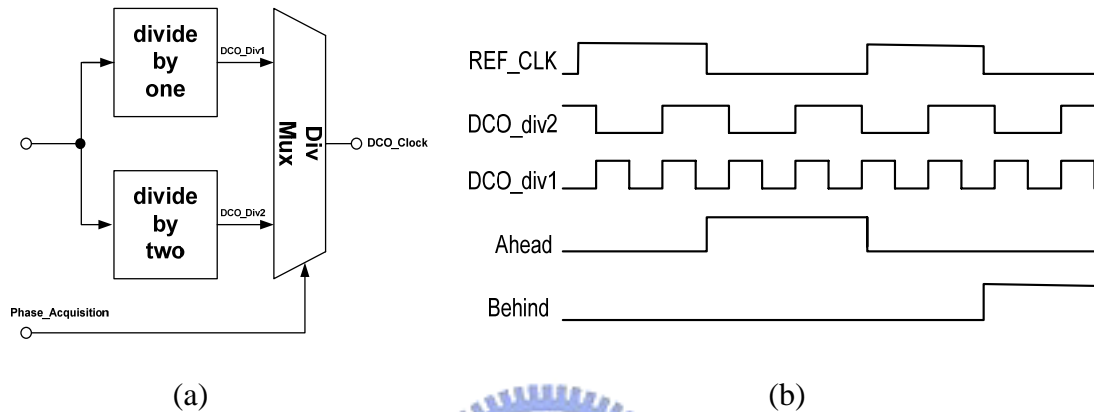


Fig.2-20 (a) Phase Acquisition Mux, (b) Phase Acquisition Resolved

There exists a pathological phase-acquisition scenario with this implementation where the false detection of a change in phase polarity can occur. The scenario arises when the initial phase error between the DCO clock and the reference clock is 180 degrees. As shown in Fig. 2-20(a), inserting a divide-by-two circuit between the DCO clock and phase detector can preclude this scenario. In addition, Fig. 2-20(b) depict a timing diagram of the resulting circuit. The mux selects the divide-by-one circuit after phase acquisition is complete, constraining phase alignment to a rising DCO clock edge.

2.5.5 Phase and Frequency Maintenance

Once the phase lock process completes, a maintenance mode begins. The ADPLL decouples this mode into phase maintenance mode and frequency maintenance mode. Phase maintenance strives to preserve the phase alignment of the DCO clock relative to the reference clock, while frequency maintenance strives to preserve the analogous match in frequency.

In phase maintenance, the ADPLL increments or decrements the DCO control word every reference cycle, based on the output of the phase detector, unless it discerns a change in phase polarity from that of the prior cycle. The value held in the phase-gain register dictates the magnitude of the changes to the DCO control word. Whenever a change in phase polarity occurs, the ADPLL transfers the anchor register contents to the DCO control register, restoring the baseline frequency.

However, reference-clock frequency drift or DCO frequency drift induced by voltage or temperature variations requires that the ADPLL has the capability of changing the baseline frequency. Frequency maintenance mode provides such means by updating the anchor register.

Chapter 3

Digitally Controlled Oscillator

Digitally controlled oscillator (DCO) is the key component of ADPLL. Like most voltage controlled oscillators (VCO), DCO consists of a frequency-control mechanism with an oscillator block. In this chapter, first, we will introduce the basic concepts and approaches, as well as some examples in DCO circuit design. Finally, the proposed DCO circuit design in our ADPLL will be presented.



3.1 Basic Concepts of DCO

The basic transfer function of DCO is as follows [22]:

$$f_{DCO}(D) = f(d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \dots + d_12^1 + d_02^0) \quad (3.1)$$

The output wave of DCO, typically in the form of square wave, which has a oscillation frequency of f_{DCO} that is a function of a digital input D . The DCO transfer function is usually defined so that the frequency f_{DCO} is changed linearly with its input word (D), hence it is also typically expressed as:

$$f_{DCO}(D) = f_{offset} + D \cdot \Delta f \quad (3.2)$$

where f_{offset} is a constant offset frequency and Δf is its frequency quantization step.

In addition, because the time period of DCO, $T_{DCO}(D) = 1/f_{DCO}(D)$, is a function

of the quantized digital input D , the transfer characteristic of DCO is evidently discontinuous. In other words, as shown in equation 3.2, this will result in a finite frequency step size Δf and hence set some fundamental limits on the achievable jitter of the ADPLL. For this reason, it is the most important that the resolution of DCO have to be sufficiently high to maintain acceptable jitter. Besides, there are still some important issues on DCO design, such as: the DCO had better approach a monotonic response to the DCO control word and own high noise immunity at the same time, so the output clock will not induce larger jitter.

3.2 Different Approaches for DCO design

As shown in Fig. 3-1, a straightforward idea to implement DCO in [19] is utilized a digital-to-analog converter (DAC) and a conventional voltage controlled oscillator (VCO). However, it is very difficult to design a high resolution DAC and the area cost will be very high due to the DAC and VCO. Besides, the VCO is an analog block making it easily be influence by power and substrate noise.

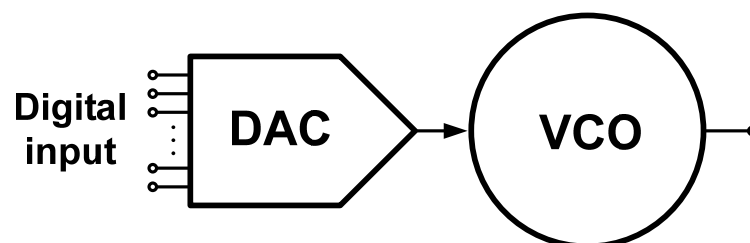


Fig.3-1 DCO composed of a DAC and a VCO

A high frequency oscillator with a program frequency divider is commonly used for another type of DCO. As shown in Fig. 3-2, the program divider receives an n -bit digital control word D which indicates the divide ratios. The DCO output clock is to be divided from a high frequency oscillator. In this arrangement, however, the DCO output frequency resolution is limited by the high frequency oscillator. In other words, this operation will require the oscillator operating at a very high frequency and hence consume much power dissipation.

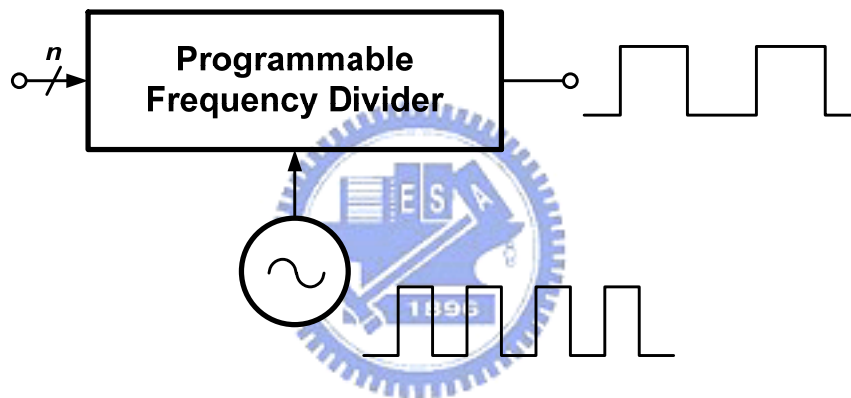


Fig.3-2 DCO composed of a high frequency oscillator and a divider

Because of the limitation on speed, directly synthesis a signal rather than dividing from a high frequency oscillator is often used in the conventional DCO. For this reason, the ring oscillator-based DCO has thus been used commonly in ADPLL for many applications today. Therefore, we will focus on the ring oscillator-based DCO design in the following.

So far, there are two main parameters to modulate the frequency of the ring oscillator. One is the total number of the delay elements, usually taken for the coarse tune method, and the other is the propagation delay time of the delay elements (i.e.

inverters), which is usually taken for the fine tune method. The first frequency modulating parameter is usually realized by a path-selection approach, and Fig. 3-3 shows the example [22]. In this example, 2^n delay buffer are connected in series. A decoder decodes an n-bit control word D into 2^n control lines. Hence, if the propagation delay time of each buffer stage is T_{buffer} , then the time resolution is $2 \cdot T_{\text{buffer}}$.

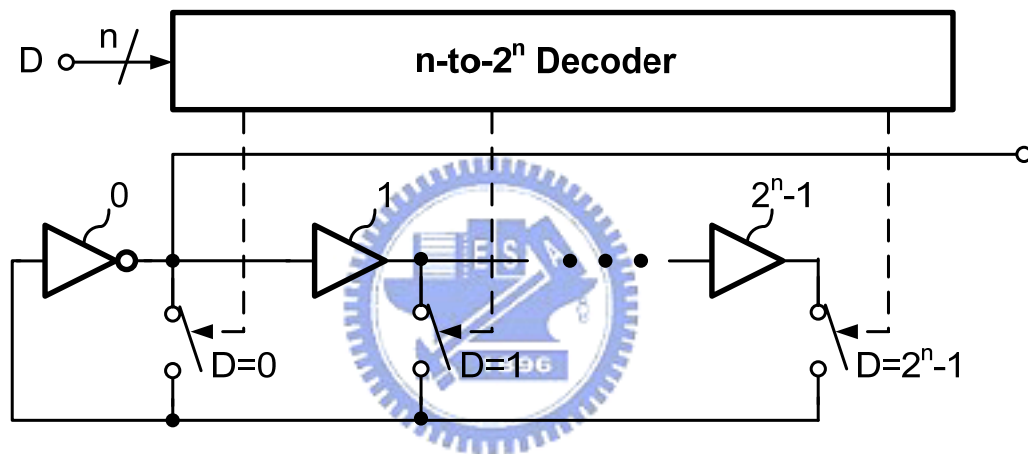


Fig.3-3 DCO realized by a path-selection method

The second frequency modulating parameter is usually realized by a digitally controlled delay element (DCDE). Moreover, we use a new DCDE in the DCO design and it has features of its monotonicity and insensitivity to PVT variations. In the following section, we will introduce several kinds of DCDE and the new DCDE circuit design.

3.3 Digitally Controlled Delay Element

There are several different architectures that have been used to implement a digitally controlled delay element (DCDE). However, they can generally be classified into the parallel-inverter-based and the single-inverter-based delay elements, individually. First, we take the parallel-inverter-based DCDEs into consideration, which are summarized in [22].

One simple DCO consists of a bank of tri-state inverter buffers was proposed in [3], [20], [21], as shown in Fig. 3-4. By enabling the numbers of tri-state inverter buffers, we can control the resolution of DCO. It is simple and easy to implement; however, it needs large area and high power dissipation for the fine tune necessarily in the DCO design. Besides, the resolution is hard to be uniform.

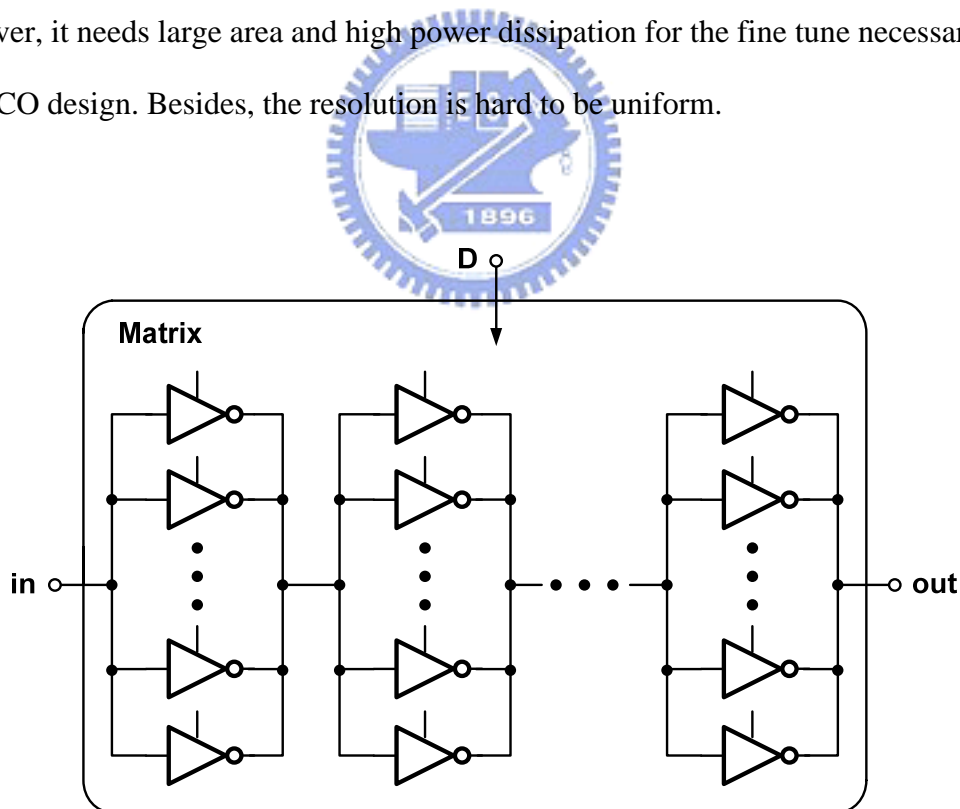


Fig.3-4 DCDE composed of a tri-buffer matrix

The other example, as shown in Fig. 3-5, a DCO is implemented by an add-or-inverter (AOI) cell and or-and-inverter (OAI) cell with two parallel tri-state inverters was proposed in [23]. The basic method is to adjust the driving capability with resistance control. The advantage is that this fine tune method of DCO cell has less area and power dissipation compared with [3], [20], [21]. However, since it's based on AOI-OAI cell to change the delay resolution, the resolution step is also hard to be uniform and sensitive to power-supply variation. Besides, is also requires an additional decoder for mapping the control input of AOI-OAI cell.

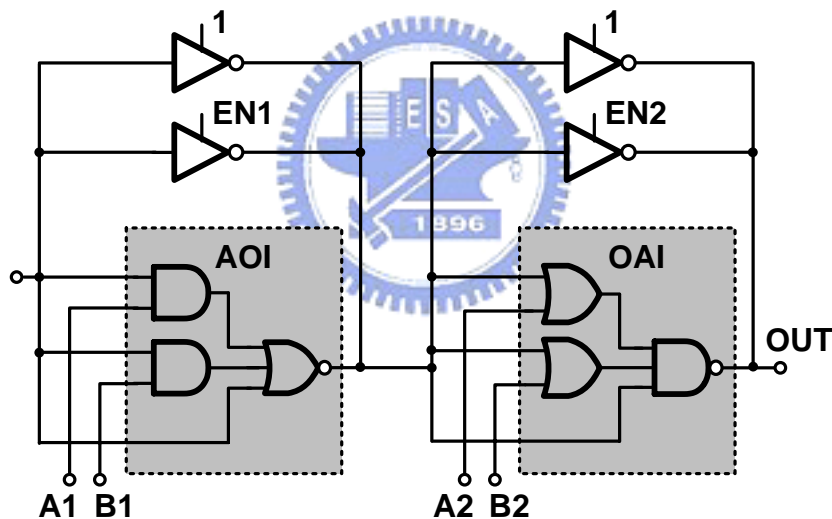


Fig.3-5 DCDE composed of an AOI-OAI

Moreover, we will keep on discussing the single-inverter-based DCDEs. Within most of the architectures, usually, a switch network of nMOS transistors is placed at the source of the nMOS transistor in a CMOS inverter, as shown in Fig. 3-6. In this circuit only the delay of the falling edge of the output voltage can be controlled by the input vector. In order to control the delay of the output rising edge, another, similar

switch network of pMOS transistors should be placed at the source of the pMOS transistor (M2) in the inverter.

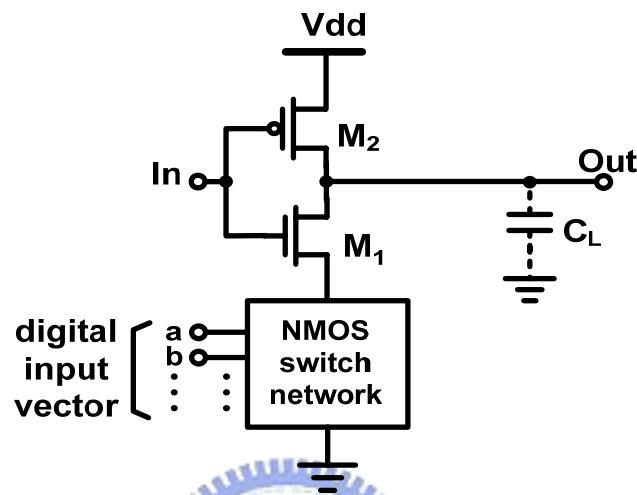


Fig.3-6 Basic structure of a delay element

The number of nMOS transistors in the switch network depends on the desired number of different separate delays and the required delay resolution. Depending upon the digital input vector, the equivalent resistor of the switch network (or the current passing through it) changes and causes the delay of the inverter to change [5], [13].

One of the main drawbacks of these delay elements is that the delay of the circuit may not change monotonically with respect to the input vector. It makes the design of the circuit more difficult, hence the circuit should be thoroughly simulated for all the possible combinations of the input vector. For example, in the case of the circuit used in [13], finding the sizes of the transistors in the switching network is a matter of optimal coding.

Fig. 3-7 illustrates a DCDE based on the current-starved inverter. The charging and discharging currents of the output capacitance (C_{L1}) of the inverter, composed of M_1 and M_2 , are controlled by two sets of current-controlling nMOS (M_{n0}, M_{n1}, \dots) and pMOS (M_{p0}, M_{p1}, \dots) transistors at the source of M_1 and M_2 , respectively. The current controlling transistors are sized in a binary fashion. It allows us to achieve binary incremental delays. As can be seen, by applying a specific binary vector to the controlling transistors, a combination of transistors is turned on at the sources of M_1 and M_2 transistors. Such an arrangement controls the rise time and fall time, and hence the delay, of the output voltage of the inverter.

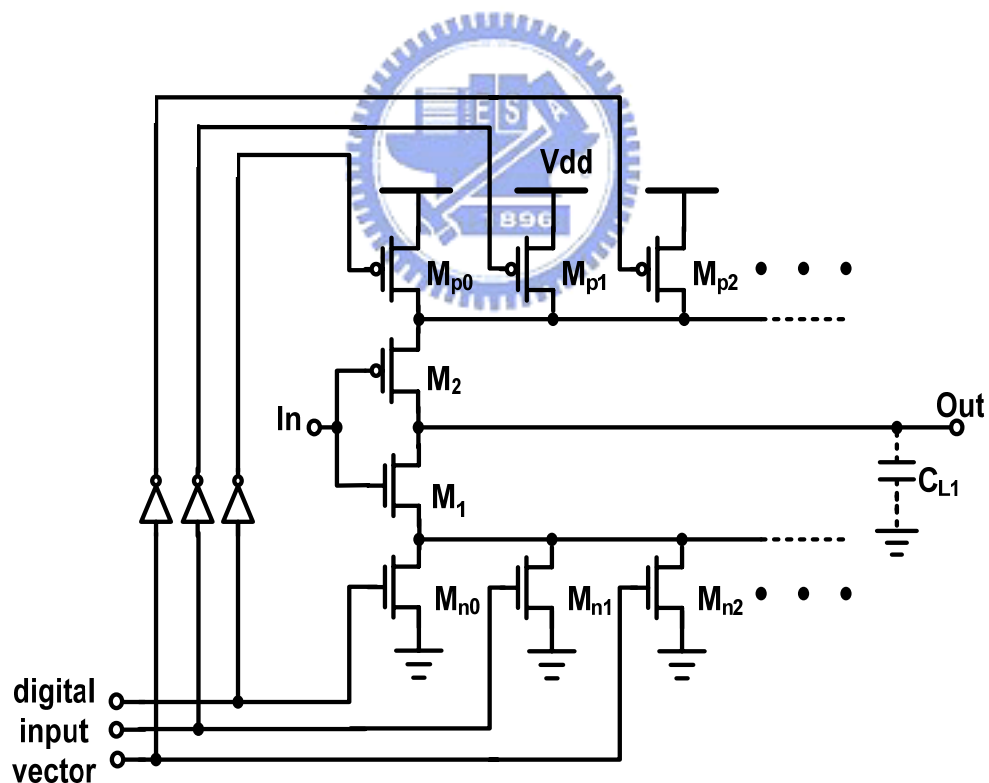


Fig.3-7 Current-starved delay element

Fig. 3-8 illustrates another technique for implementing a DCDE. In this circuit, a variable resistor is used to control the delay. A stack of n rows by m columns of nMOS transistors is used to make the variable resistor. This resistor subsequently controls the delay of M_1 . In the circuit of Fig. 3-8, only the falling edge of the Out can be changed with the input vector. Similarly, another stack of pMOS transistors can be used at the source of the pMOS transistor, M_2 , to have control over the delay of the rising edge.

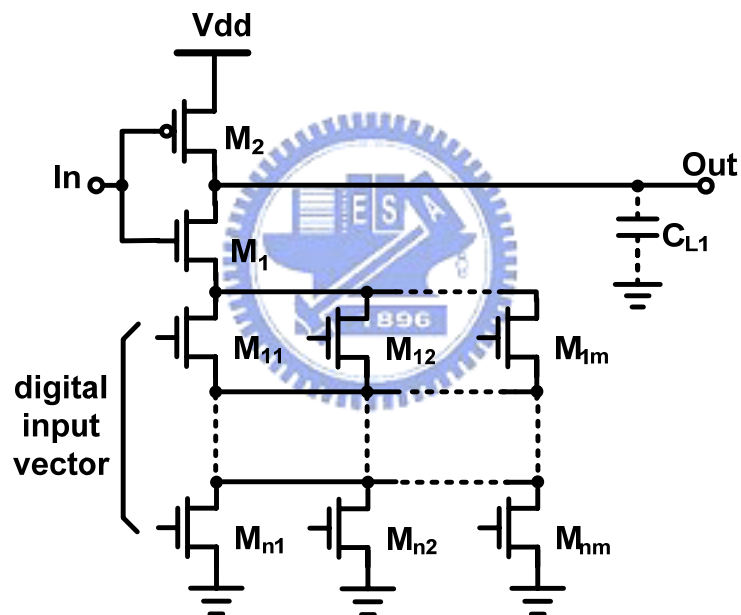


Fig.3-8 Another delay element

One of the problems with the above mentioned single-inverter-based DCDE architectures is the nonmonotonic delay behavior with ascending binary input pattern. As can be seen in the circuits of Figs. 3-7 and 3-8, the input vector changes the effective resistance of transistor(s) placed at the source of the nMOS or pMOS transistors of the inverter. This not only changes the resistance at the source of M_1 or

M_2 , but also changes the parasitic capacitance associated with transistors at these nodes. This is because the parasitic capacitance at the drain of a MOSFET is different in the *ON* and *OFF* states. Therefore, there are two factors depending on the input vector to affect the delay:

(1) The resistance of the controlling transistors:

The circuit delay can be increased/decreased by increasing/decreasing the effective *ON* resistance of the controlling transistors at the source of M_1 (M_2).

(2) The effective parasitic capacitance of the controlling transistors:

As the effective capacitance of the controlling transistors at the source of M_1 (M_2) increases due to the input vector, the charge sharing effect causes the capacitance at the output of the current-starved inverter to be (dis)charged faster and the overall delay of the circuit decreases.

Because the W/L ratio of the controlling transistors have to change in binary fashion, usually, the channel length L , is thus increased to realize a small W/L ratio. A longer transistor puts a higher resistance and a larger parasitic capacitance at the source of M_1 (M_2). A larger resistance increases the delay; however, a larger parasitic capacitance decreases the delay. Therefore, it may make monotonic characteristic of the DCDE can not be ensured with ascending input vector. This situation will be further complicated as the number of delay controlling transistors increases.

For this reason, it becomes difficult to predict the circuit delay for a given input vector and will cause the circuit to be simulated for all the possible input combinations during the design phase. The design of high-resolution delay element becomes a nontrivial task due to the lack of a one-to-one relationship between transistor sizes and corresponding delay. If, in the design phase, the desired delays are

controlling current will later be mirrored to M_8 and M_{11} , and controls the delay of the inverter. Note that transistor M_p is always on. This circuit can implement 2^N different delays where N is the number of pMOS controlling transistors. Note that the parasitic capacitances at the source of M_9 and M_{10} are the same for all the input vector combinations.

Therefore, when the input vector changes, only the (dis)charging current of the inverter changes, and the charge sharing remains the same. This causes the delay of the circuit to change monotonically with respect to the input vector, which makes the design of this circuit straightforward compared to the other delay elements.

Another point which is worth mentioning is that both the rising and falling edge delays can be varied by this circuit. This has come at the expense of three more transistors (M_6, M_7 , and M_{11}), while in the conventional delay elements, the number of added transistors for this purpose is more. Note that transistors M_6, M_7 , and M_{11} do not need to be very large, while the delay-controlling transistors in conventional delay elements are large and consume extra area due to their binary sizing scheme.

The design procedures of the new DCDE are explained as follows [15]:

(1) Transistor M_8 / M_{11} should be much smaller than M_9 / M_{10} such that the discharging current is controlled by M_8 / M_{11} . The ratio of transistors M_{10} and M_9 should be μ_n / μ_p where μ_n and μ_p represent electron and hole mobilities. Transistor (M_5, M_6) / M_7 can be the same size as M_8 / M_{11} , since these transistors make the current mirrors. However, these transistors may have different sizes to reduce the static power consumption of the DCDE, as explained in [14].

(2) The number of pMOS controlling transistors (N) can be obtained from the desired number of different delays (m) of DCDE such that $m=2^N$. Moreover, the circuit must

contain one more pMOS transistor (M_4) which is always on. In our case, we have selected 7+1 pMOS controlling transistors, which provide us 128 different delays.

(3) Assuming transistors M_1 to M_7 are not present, transistor M_p is sized to get the maximum desired delay.

(4) After sizing M_p , we put another pMOS transistor (e.g., M_0) in parallel to M_p to obtain the minimum desired delay. Note that M_0 is not shown in the figure since this transistor is subsequently fragmented into N (7, in our case) smaller transistors.

(5) Transistor M_0 is now fragmented into $N=7$ transistors, (M_1 to M_7), in a binary fashion. That is:

$$(W/L)_{M_i} = \frac{2^{i-1}}{2^7 - 1} (W/L)_{M_0}, \quad i = 1, 2, \dots, 7 \quad (3.1)$$



3.4 Digitally Controlled Oscillator Architecture

The architecture of DCO is presented in Fig. 3-10. The DCO is composed of eight Digitally Controlled Delay Elements (DCDE), several transmission gates, balance inverters, output driving inverters, and one controlling NAND gate to enable the DCO. The proposed DCO circuit has total 11-bit resolution, including coarse tune and fine tune parts. In coarse tune part, the number of delay element is chosen to cover different frequency band. The number of delay element will increase rapidly through the increasing path-selecting control bits, thus, the circuit will consume much more power. Furthermore, the cover band between band to band will also decrease the operational frequency range. In a nutshell, only 1-bit control word has been used for path selector. This selection occurs based on the first frequency comparison at the

beginning of the phase lock.

In fine tune part, the new DCDE [14], [15] is adopted with 7-bit resolution in consideration of linearity. As mentioned above, the delay of the DCDE changes monotonically with respect to the digital input vector. The main idea of the DCDE is to adjust the delay difference by using current mirror-based circuit in binary-weighted fashion. The most significant advantages of such delay element are its monotonic characteristic and PVT variations immunity.

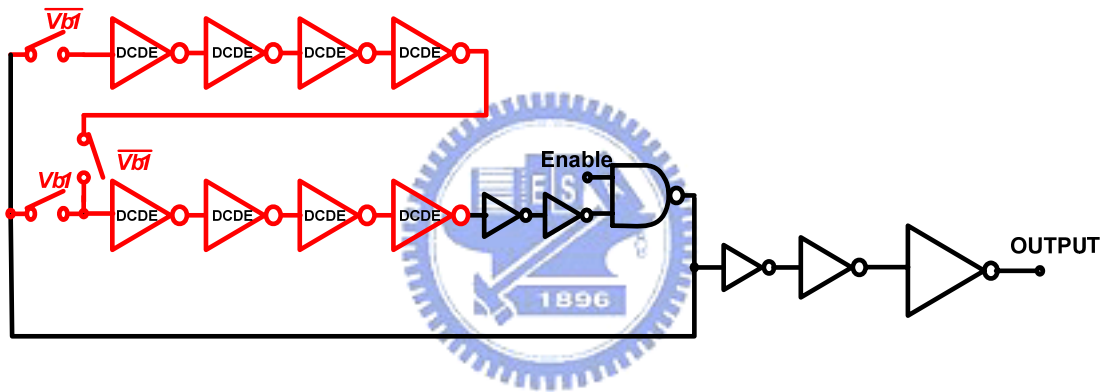


Fig.3-10 The proposed DCO architecture

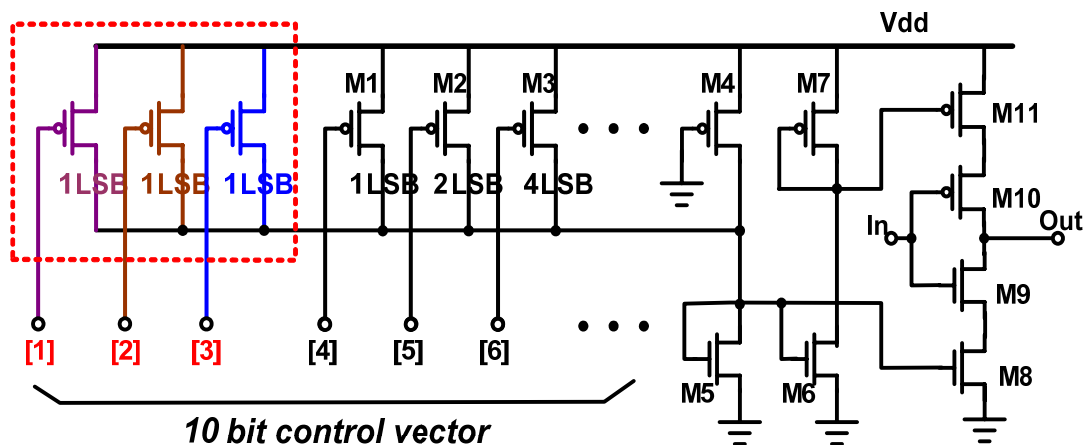


Fig.3-11 The Modified DCDE

The minimum delay difference caused by an LSB, which corresponds to the smallest control device in the DCDE. Thus, when the LSB asserts, the smallest control device is *ON* in each DCDE. However, the minimum delay difference would increase in proportion to the increase of delay element. Based on that, a modified DCDE to extend the resolution is presented, shown in Fig. 3-11. Initially, we could take every two DCDE as a group, and choose one of them to add a smallest control pMOS. In other words, the control bit 3 affects four DCDEs and higher order bits affect eight DCDEs. Hence, it will increase extra 1-bit resolution of the DCO circuit. Similarly, two more bits of resolution result from using the same pMOS device in only two of eight and one of eight DCDEs respectively. To sum up, 10-bit resolution is achieved by the fine tune part of proposed DCO.

The simulation result is shown in Fig. 3-12. Table 3-1 shows frequency ranges of the two modes of DCO. The total operation frequency range is in 260 MHz – 1.15 GHz, and the LSB resolution is 0.4 – 20 ps. Power consumption is 0.9 mW at 1 GHz.

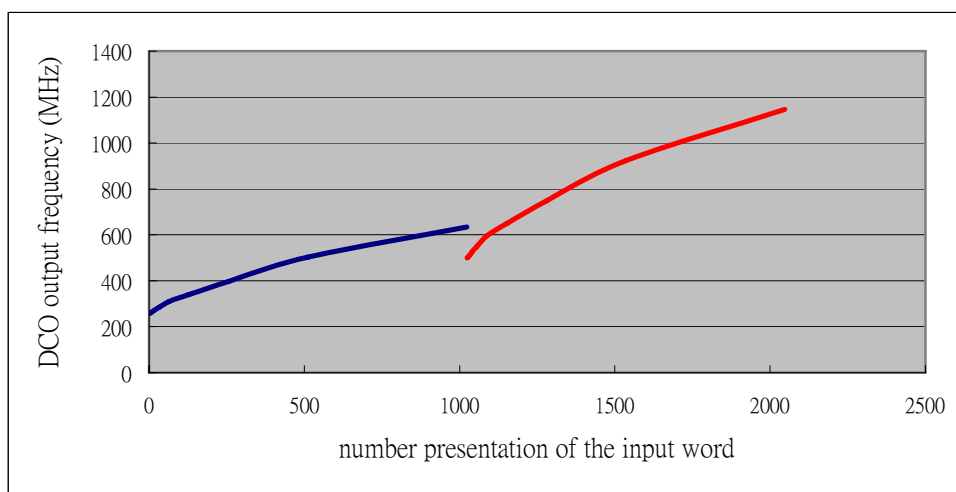


Fig.3-12 DCO Frequency V.S. number of the input vector

Table 3-1 Frequency Range of DCO with Different Environments

<i>The Mode of DCO</i>	<i>Frequency Range (MHz)</i>		
	<i>25°C 1.2V</i>	<i>25°C 1.1V</i>	<i>75°C 1.2V</i>
<i>High Frequency Mode</i>	500 – 1150	465 – 1040	454 – 1060
<i>Low Frequency Mode</i>	260 – 635	240 – 578	233 – 588



Chapter 4

A Low Power ADPLL Circuit Design

In this chapter, we will introduce the proposed ADPLL circuit design. The ADPLL has gained increased attention in recent years. All analog building blocks are replaced with digital representations in ADPLL. The term “all-digital PLL” is used for a particular reasons: all signals within this PLL are digital values; no analog signal is used.



4.1 Architecture of The ADPLL

As mentioned above, the conventional ADPLL uses four loosely coupled modes of operation: frequency acquisition, phase acquisition, phase maintenance, and frequency maintenance [5], [6]. However, by using a modified PFD architecture in the ADPLL design, we can combine the frequency acquisition and phase acquisition modes in the conventional ADPLL to the frequency/phase acquisition mode, and also combine the frequency maintenance and phase maintenance modes to the frequency/phase maintenance mode in our ADPLL design.

Fig. 4-1 depicts a block diagram of the ADPLL. The DCO control register in the control unit holds the 11 b, binary weighted DCO control word, which dictates the frequency of the DCO. Arithmetically incrementing or decrementing the DCO control

word modulates DCO frequency and phase. The frequency-gain register or phase-gain register in the control unit will provide operands to the adder/subtractor, hence the adder/subtractor can provide the updates to the DCO control register.

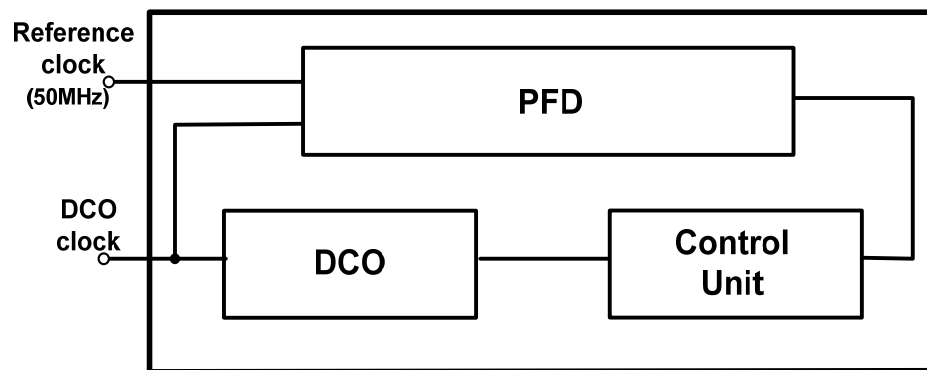


Fig.4-1 The ADPLL Block Diagram

Frequency/phase maintenance mode begins with frequency/phase acquisition mode. The goal of this mode is to lock frequency and phase of the DCO to that of the match-delay reference clock. In this mode, a modified binary-search algorithm sweeps the frequency range of the output of DCO counter in the PFD to match that of the reference clock. The search algorithm which has been introduced in previous chapter is also shown in Fig. 4-2. It makes incremental changes to the DCO control word based on the output of the PFD. The value held in the frequency-gain register determines the magnitude of the changes.

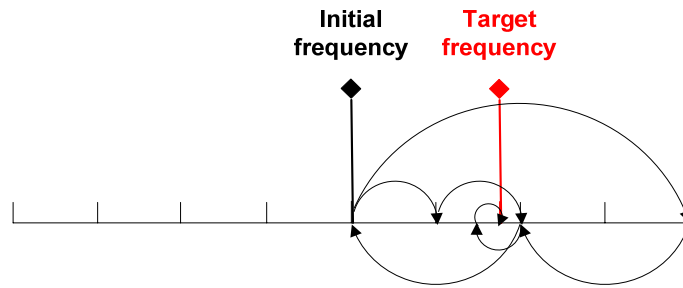


Fig.4-2 The Modified Binary Search

After the frequency/phase acquisition is complete, in other words, the ADPLL enters the lock state, the frequency or phase of the reference clock and that of the DCO output would still be changed by the PVT variations. For this reason, the system will enter the frequency/phase maintenance mode to make sure of the phase error being under control. In this mode, the value held in the phase-gain register determines the magnitude of the changes, and we will use another algorithm to adjust the phase-gain value, which will be introduced in the section of Control Unit later.

4.2 Circuit Design of The ADPLL

4.2.1 Phase/Frequency Detector

In the conventional ADPLL design, PFD is composed of frequency comparator and phase detector. The frequency comparator (FC) accepts the reference clock and the DCO output as its inputs. By these two signals, the FC generates FAST, SLOW,

and ENABLE output signals for the DCO. The PFD uses the reference clock edge to assert the DCO Enable signal, which forces the reference clock edge and the DCO output edge to align in phase. This initial phase alignment makes an accurate frequency comparison possible after one reference cycle. Then, the phase detector (PD) also uses the same two signals to generate AHEAD or BEHIND output signals.

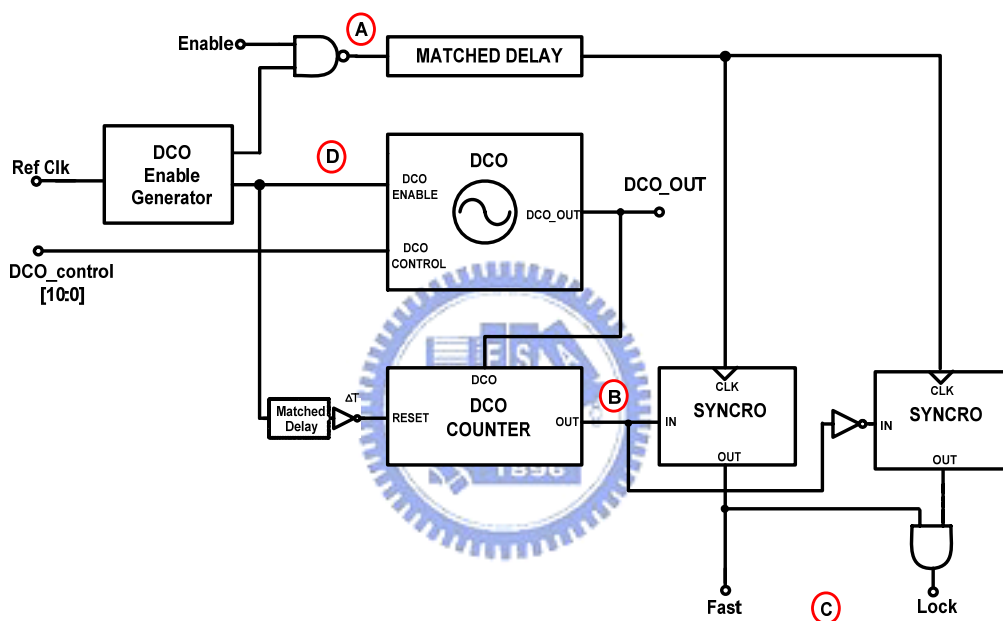


Fig.4-3 Phase/Frequency Detector

As mentioned in the chapter 2, the conventional FC takes one-half reference cycle for synchronization before asserting SLOW or FAST. In the rest clock, the DCO is disabled. That would need two reference clock cycles to finish a full FC iteration. To settle such difficulty, a dual-mode one-cycle PFD is proposed, shown is Fig. 4-3.

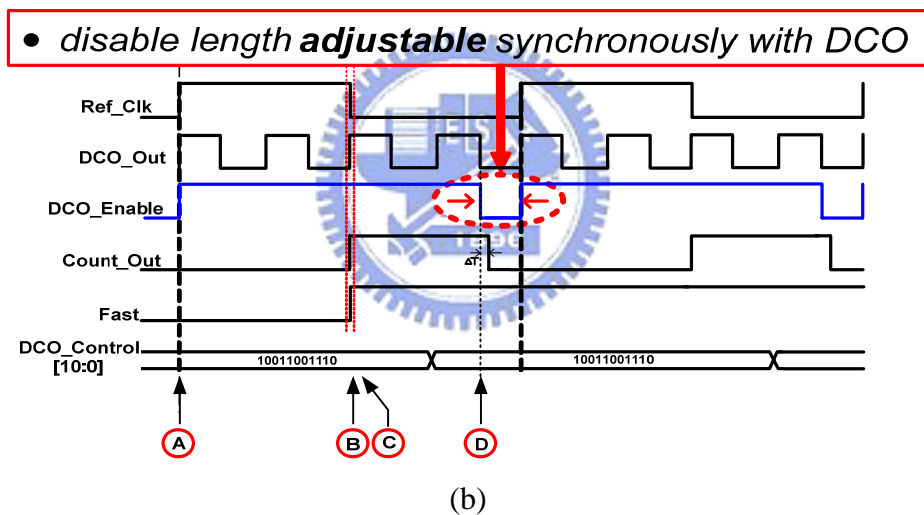
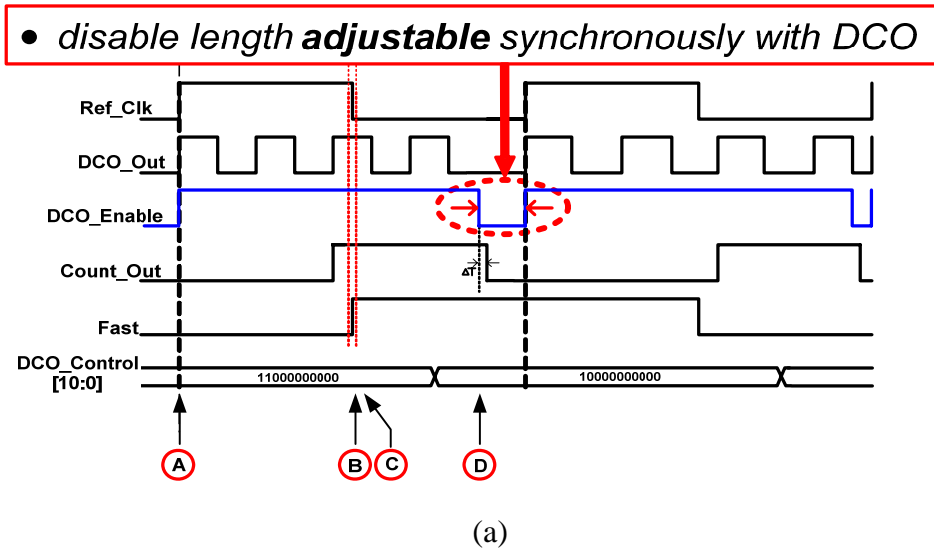


Fig.4-4 Timing diagram of the Phase/Frequency Detector

(a) unlock state, (b) lock state (for example: divider ratio=2)

Above all, let us see Fig. 4-4 (a) and (b), which show the timing diagrams of two frequency comparison iterations in the unlock state and lock state, respectively. The operation flow of the modified PFD is roughly similar to the conventional PFD, shown in Fig. 2-18. However, compared to the conventional PFD, there are some

modifications in our design so as to reduce lock cycle of the ADPLL: First, the frequency and phase detection point is set at the falling edge of the reference clock. Second, a new flag signal is generated by DCO enable generator circuit, shown in Fig. 4-5. It is realized based on a pulse generator, which is mainly composed of a delay element and a NAND gate. The DCO Enable signal will only disable the DCO before the rising edge of next reference clock cycle, and by using the replica of the DCO as the delay element, it will last half of the DCO output cycle to synchronize with the reference clock.

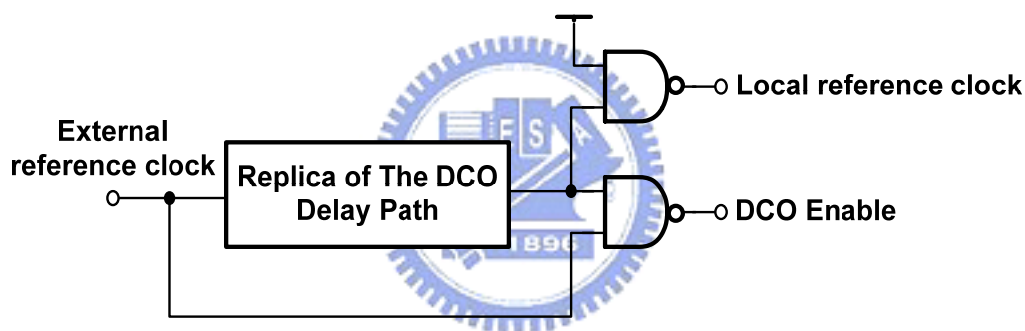
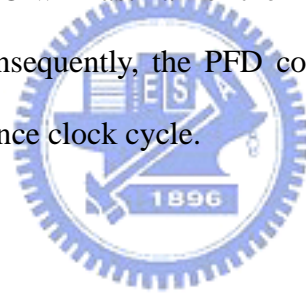


Fig.4-5 DCO Enable Generator

Then, the operation flow of the modified PFD will be introduced briefly in the following. As shown in Fig. 4-3, at first, for an accuracy frequency/phase comparison, the PFD uses the rising edge of the external reference clock to assert the DCO Enable at **A**, and the following falling edge captures the output and the inversion of DCO counter at **B**, the input to the synchronizer. Then, the synchronizer will pull up or pull down the FAST and Lock signal at **C**, depending on the early or late relation between the matched-delay reference clock and the output of DCO counter. If the output of DCO counter arrives before the falling edge of the reference clock, then the output of

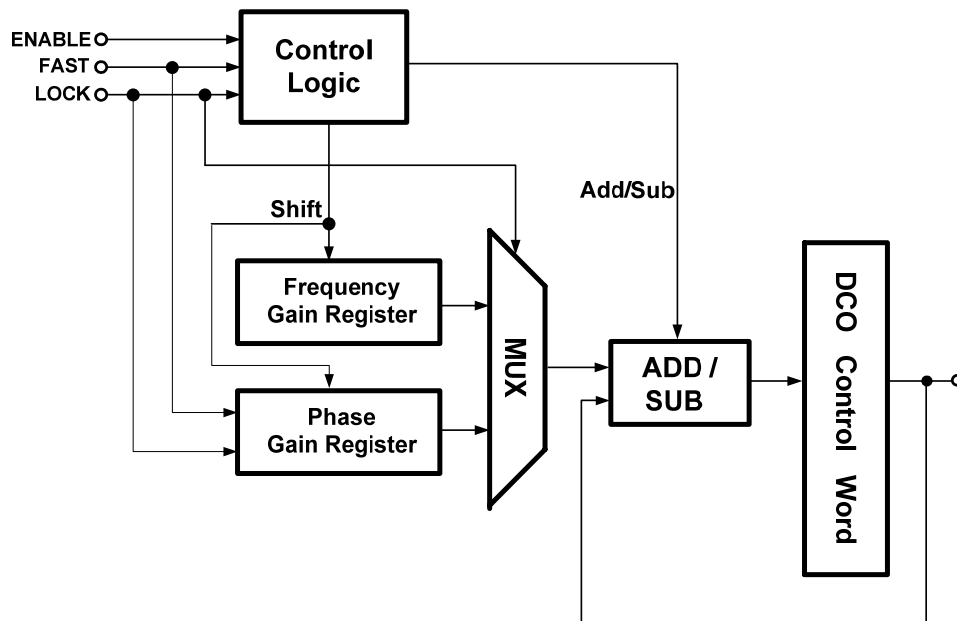
DCO counter is defined as FAST, and pulled up the FAST signal. Otherwise, it will be defined as SLOW, and pulled down the FAST signal. In addition, only when both the two synchronizers' outputs are high, the Lock signal will be asserted, and the ADPLL will enter the lock state. Afterwards, the FAST and Lock signal will be transferred to the control unit, which will be introduced in the later section.

After the control unit operates completely, it will update the control word of DCO, therefore the DCO will operate in a new oscillation frequency to trace the reference clock further. Finally, the DCO Enable generator will force the DCO to be disabled at **D**, waiting for the next comparison cycle. Furthermore, as mentioned above, the disable time of DCO will last half of the DCO output cycle to synchronize with the reference clock. Consequently, the PFD could finish frequency and phase comparison in only one reference clock cycle.

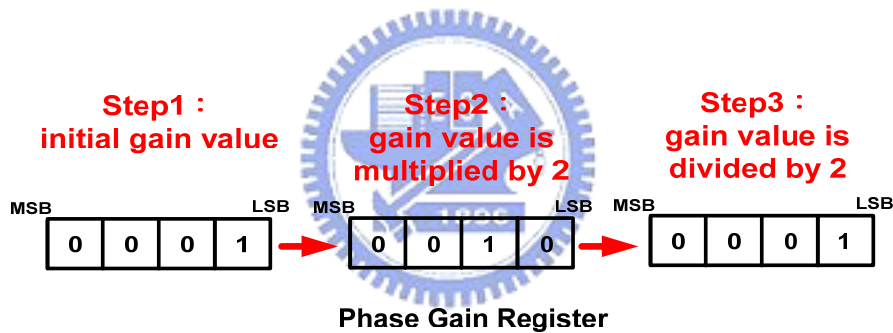


4.2.2 Control Unit

Control unit (CU) will adjust the DCO control word to change DCO output frequency according to the FAST signal and Lock signal received from PFD. Architecture of the control unit is shown in Fig. 4-6 (a), and it will adopt the modified binary search algorithm when the system is in the phase/frequency acquisition mode. In implementing the gain strategy of the binary search algorithm, an adder/subtractor receive both the value of a frequency gain register and the DCO control word. The frequency gain register is a 11-b, unidirectional shift register with binary weighted bits, analogous to the DCO control word.



(a)



(b)

Fig.4-6 (a) Block diagram of the Control Unit

(b) An example of the phase gain strategy

In the phase/frequency acquisition mode, first, CU will decide if it needs to shift the frequency gain word once to the right (i.e., decreases the gain by a factor of two) by telling if polarity of the FAST signal has changed. Then, the control word will be added to value in the frequency gain register (i.e., the frequency gain word) if the FAST signal is low, or it will be subtracted to the frequency gain word if the FAST

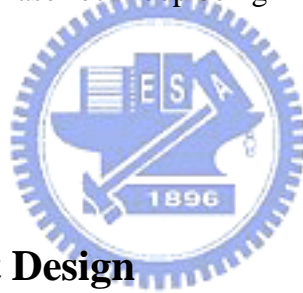
signal is asserted. In a conventional implementation [5], the add MUX receives the odd bits of the frequency gain register (the add gain), and the subtract MUX receives the even bits of the frequency gain register (the subtract gain). However, in our design, the frequency gain word is used as both the add gain and subtract gain, as well as, the adder and subtracter are combined together such that it can save an adder or a subtractor circuit and also the associated logic in the muxes.

After the ADPLL is locked, the frequency of reference clock or DCO clock would still be influenced by the PVT variations, for this reason, the system have to enter the maintenance mode at this moment, the same as the conventional design [5] [6]. Additionally, as mentioned above in this chapter, the phase maintenance and frequency maintenance modes in the conventional ADPLL are also combined to a phase/frequency maintenance mode in our design. Hence, the goal of this mode is to preserve the analogous match in frequency and the phase alignment of the DCO clock relative to the reference clock at the same time.

Similar to the operation of the phase/frequency acquisition mode, based on the PFD output, the ADPLL increments or decrements the DCO control word every reference cycle in the phase/frequency maintenance mode. However, the difference to the phase/frequency acquisition mode is that the magnitude of the changes to the DCO control word (i.e., the gain value) has changed to the value held in the phase gain register. Besides, in the maintenance mode, the ADPLL adopts a different gain strategy to adjust the gain value (the phase gain strategy). While the phase gain register still uses the bit-shifted gain technique of the acquisition mode, it now employs a variable shift displacement. This modification gives the ADPLL increased flexibility to reduce gain for improved DCO output jitter or increase gain for

improved phase and frequency tracking in the presence of drift.

Fig. 4-6 (b) shows an example of the procedure of gain value adjustment. The step 1. is to set the initial value of the 4-bit phase gain word as 1. The phase gain register will be shifted once (i.e., increases the phase gain value by a factor of 2) to the left whenever the control unit detects polarity of the FAST signal from the PFD remains the same for eight successive reference cycles. The register will be shifted once to the right (i.e., decreases the phase gain value by a factor of 2), otherwise, whenever the polarity of the FAST signal has changed. In conclusion, the gain register can only be 0001, 0010, 0100, and 1000. Therefore, the phase gain strategy can make sure of the phase error of the phase-lock loop being minimized.



4.2.4 The Sub-Circuit Design

(1) D Flip-Flop:

In the ADPLL, D Flip-Flop (DFF) have been used in many circuits, such as: the DCO counter, the synchronizers of PFD, the frequency gain register, the phase gain register, and the DCO control register. The DFFs are all realized by the true single-phase clocked circuits (TSPC), shown in Fig. 4-7, because of its good performance well-known at high frequency.

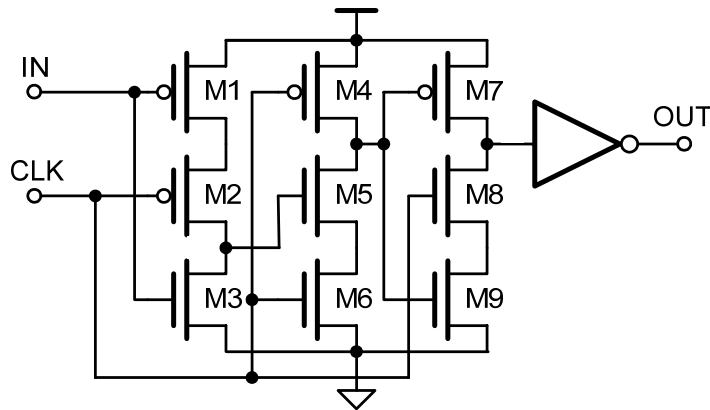
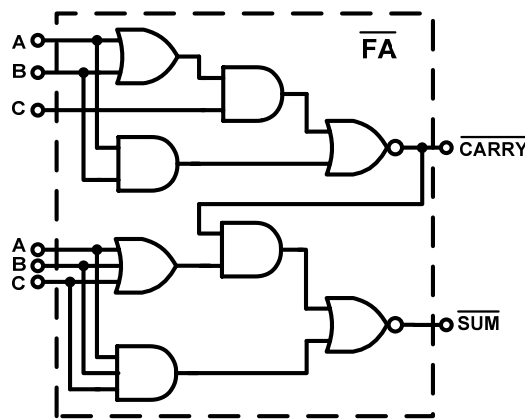


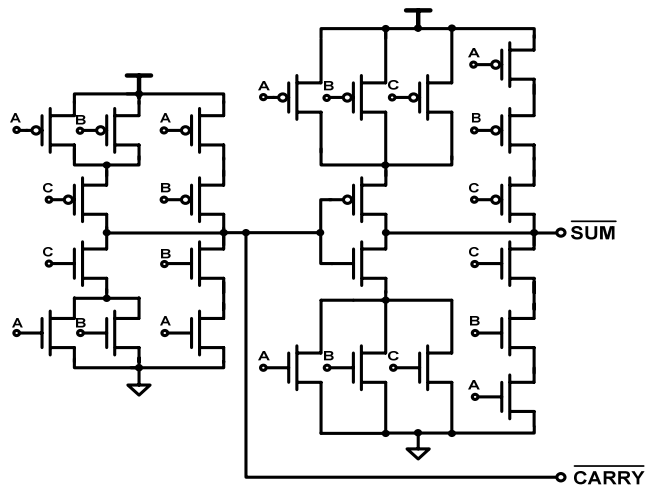
Fig.4-7 TSPC DFF

(2) Adder / Subtractor:

In the ADPLL design, there is roughly half of the reference cycle (i.e., 10nsec) for full operation of the control unit, therefore, for the power and area consideration, we can just use a 11-bit modified ripple adder/subtractor, which is shown in Fig. 4-9. In addition, the logical diagram and the transistor-level circuit of 1-bit full adder (FA) is also shown in Fig. 4-8 (a) and (b), respectively.



(a)



(b)

Fig.4-8 1-bit \overline{FA} (a) logic diagram, (b) transistor-level circuit

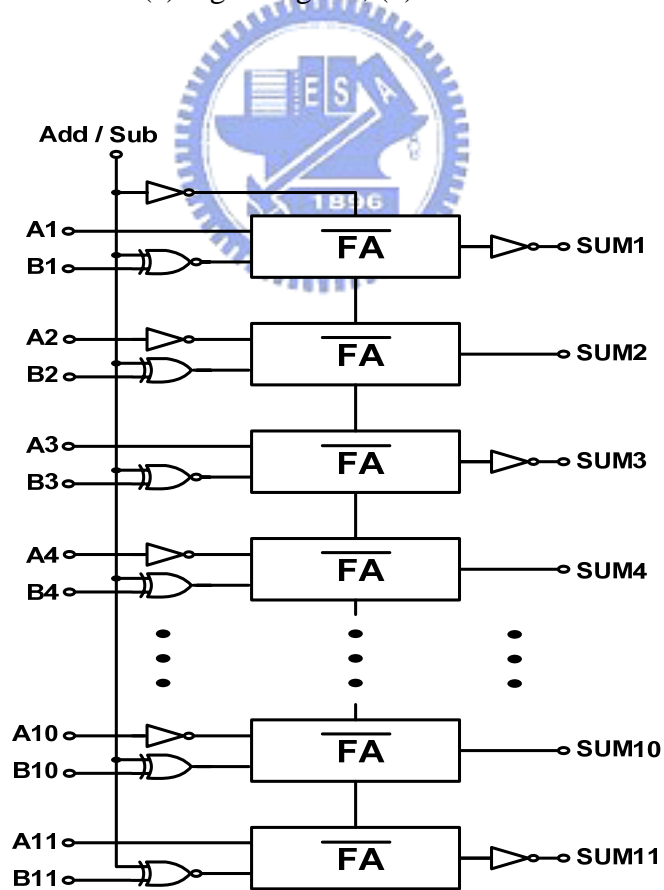


Fig.4-9 11-bit modified ripple adder/subtractor

Chapter 5

The Implementation of Frequency Synthesizer

In this chapter, at first, we will introduce the fundamentals of frequency synthesizer. Then, the ADPLL-based frequency synthesizer based on the adjustable counter length mechanism will be presented. Finally, it also shows the implementation of layout and the simulation results.



5.1 Frequency Synthesizer Architecture

As mentioned in the chapter 1, synthesizers often require that the output frequency of a PLL be a multiple of the input frequency. The high accuracy for the different output frequency often mandates the use of PLLs in synthesizers because under locked condition, the output frequency of a PLL bears an exact relationship with the input frequency. In this section, we will introduce several architectures for the frequency synthesizer [16].

5.1.1 Integer-N Architecture

Depicted in Fig. 1-2, as mentioned above, such a topology produces $f_{out} = M \cdot f_{REF}$, where M (i.e., modulus) varies in unity steps from M_L to M_H . The frequency divider employed in Fig. 1-2 must provide a variable modulus given by $M = M_L + k$, $k = 0, 1, \dots, N$. An example of such a circuit is a “pulse-swallow divider,” illustrated in Fig. 5-1. The divider consists of a “prescaler,” a “program counter,” and a “swallow counter.” We briefly describe the operation of the circuit here. Let us first make three observations: (1) the prescaler divides the input by either $N + 1$ or N according to the logical state of the modulus control line, (2) the program counter always divides the prescaler output by P , and (3) the swallow counter divides the prescaler output by S , where S is determined by the digital input and can vary from 1 to the maximum number of channels. This counter also has a reset input. We will show that $f_{out} = f_{in} / (NP + S)$.

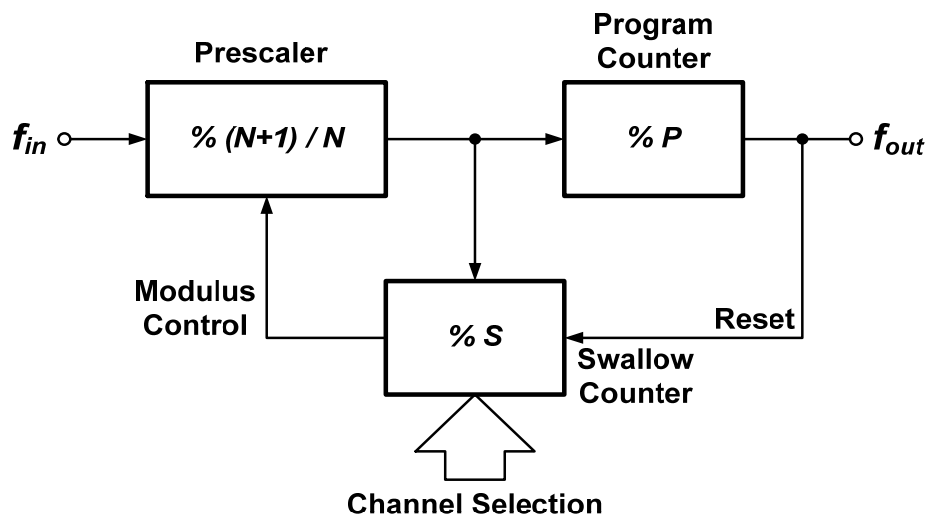


Fig.5-1 Pulse swallow frequency divider

When the circuit begins from the reset state, the prescaler divides by $N + 1$. The prescaler output is divided by both the program counter and the swallow counter until the latter is “full,” i.e., it has counted S pulses. At this point, that is, after $(N + 1)S$ cycles at the main input, the swallow counter changes the state of the modulus control line, making the prescaler divide f_{in} by N . Note that before this change, the program counter has sensed a total of S pulses. After the modulus changes, the prescaler and program counter continue to divide until the latter is full. Since the program has already sensed S pulses, it requires $P - S$ cycles at its input, and hence $(P - S)N$ pulses at the main input, to reach overflow. Thus, the output generates one complete cycle for every $(N + 1)S + (P - S)N = PN + S$ cycles at the input. The operation repeats after the swallow counter is reset.

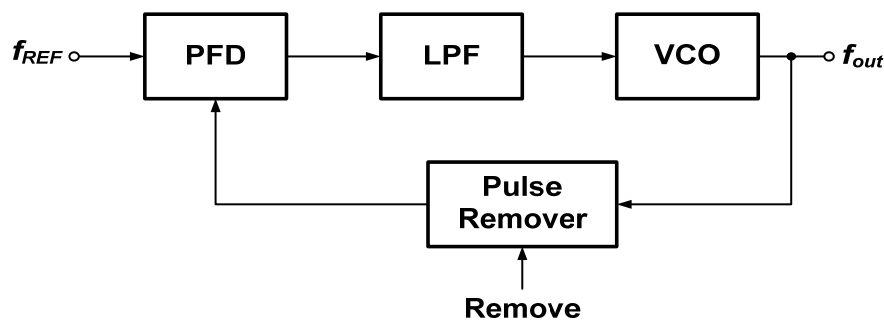
The simplicity of the integer- N architecture has made it a popular choice for many decades. In RF systems, the synthesizer has commonly been partitioned into three separate chips: the VCO; the dual-modulus prescaler; and the combination of the program counter, the swallow counter, the PFD, and the charge pump. As the fast parts of the system, the VCO and the prescaler has typically been fabricated in silicon bipolar or GaAs technologies and the rest in CMOS technology. Note that a buffer is usually interposed between the VCO and the prescaler to isolate the former from the switching noise in the latter.

5.1.2 Fractional-N Architecture

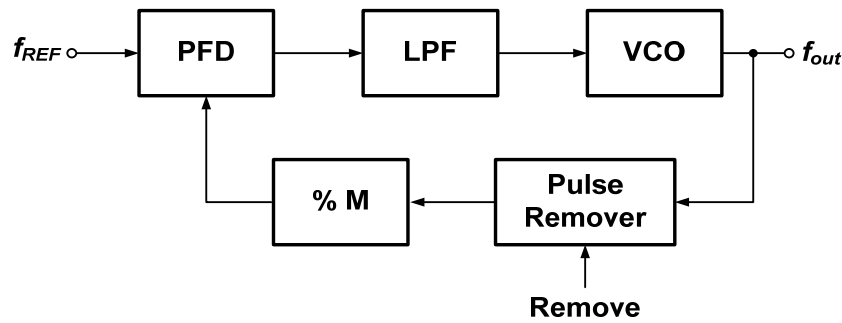
In the integer- N architecture, the loop bandwidth is limited because the input

reference frequency must be equal to the channel spacing. This, in turn, results from the property that the output frequency changes by only integer multiples of f_{REF} . In “fractional- N ” synthesizers, on the other hand, the output frequency can vary by a fraction of the input frequency, allowing the latter to be much greater than the channel spacing.

Fig. 5-2 (a) shows a simple phase-locked fractional- N architecture. In addition to the PFD, LPF, and VCO, the loop incorporates a pulse remover, a circuit that blocks one input pulse upon assertion of the remove command. Since under locked condition, the two frequencies presented to the phase detector must be equal, the average output frequency of the pulse remover equals f_{REF} , and hence $f_{out} = f_{REF} + 1/T_p$, where $1/T_p$ is the period with which the remove command is applied. Note that f_{out} can vary by a fraction of f_{REF} because the frequency $f_p = 1/T_p$ can be derived from f_{REF} by simple division. Provided by a crystal oscillator, f_{REF} is typically limited to a few tens of megahertz. Thus, as shown in Fig. 5-2 (b), fractional- N loops incorporate a divider in the feedback to generate high output frequencies.



(a)



(b)

Fig.5-2 (a) Simple fractional-N synthesizer, (b) use of divider in the loop

While the original fractional- N topology was based on the pulse remover concept [17], modern implementations of this architecture operate on a somewhat different principle. Depicted in Fig.5-3, such a synthesizer replaces the pulse remover and the divider of Fig. 5-2 (b), with a dual-modulus prescaler. If the prescaler divides by N for A output pulses of the VCO and by $N + 1$ for B output pulses, then the equivalent divide ratio is equal to $(A + B) / [A / N + B / (N + 1)]$. This value can vary between N and $N + 1$ in fine steps by proper choice of A and B . The resulting modulus is sometimes written as $N.f$, where the dot denotes a decimal point and N and f represent the integer and fractional parts of the modulus.

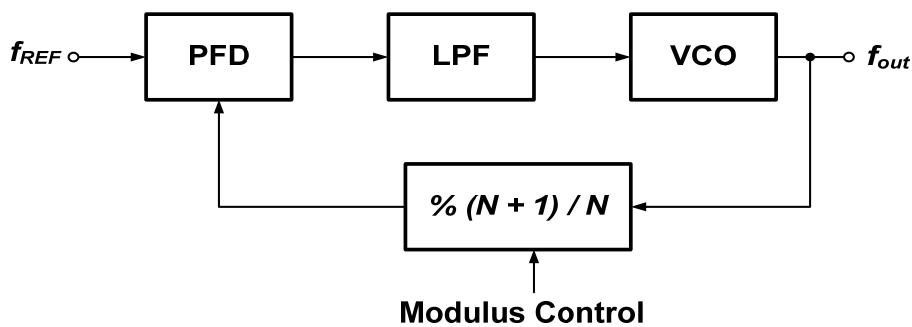


Fig.5-3 Fractional-N synthesizer using a dual-modulus divider

As an example, consider the circuit in Fig. 5-4, where $f_{REF} = 1$ MHz and $N = 10$. Let us assume the prescaler divides by 10 for 9 reference cycles and by 11 for one reference cycle. The total number of output pulses is therefore equal to $9 \times 10 + 11 = 101$, whereas the reference produces 10 pulses. In other words, the divide ratio is equal to 10.1 and $f_{out} = 10.1$ MHz.

With f_{REF} in the range of tens of megahertz, the loop bandwidth of a fractional-N synthesizer can be as high as a few megahertz, yielding a fast lock transient as well as suppressing the VCO close-in phase noise.

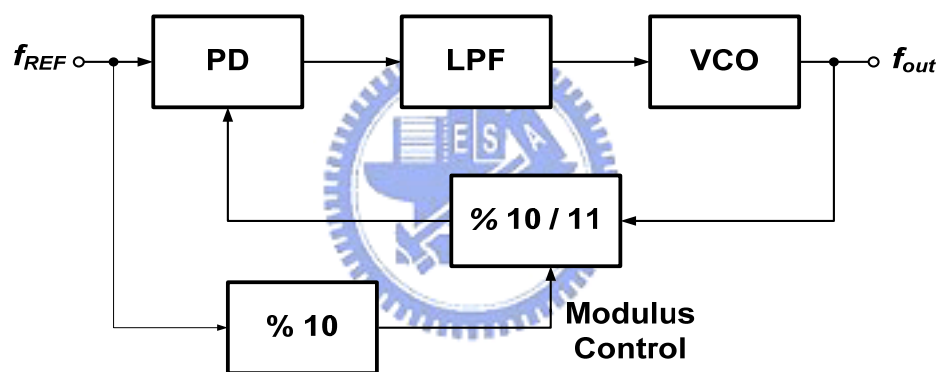


Fig.5-4 Example of a fractional-N synthesizer

5.2 Frequency Dividers

Frequency dividers are important in the study of frequency synthesizers. In addition to the issue of speed and power dissipation, the phase noise of dividers is also critical for it corrupts the feedback signal in synthesizers. In this section, we will introduce some divider topologies.

5.2.1 Divide-by-Two Circuits

As shown in Fig.5-5, a divide-by-two circuit can be realized as two latches in a negative feedback loop. Only if CK and \overline{CK} are precisely complementary and the two latches match perfectly, this configuration can provide quadrature phases at X and Y . Device mismatches typically result in phase imbalances as large as 5° . In addition, additional phase imbalances could arise if CK and \overline{CK} are not differential exactly.

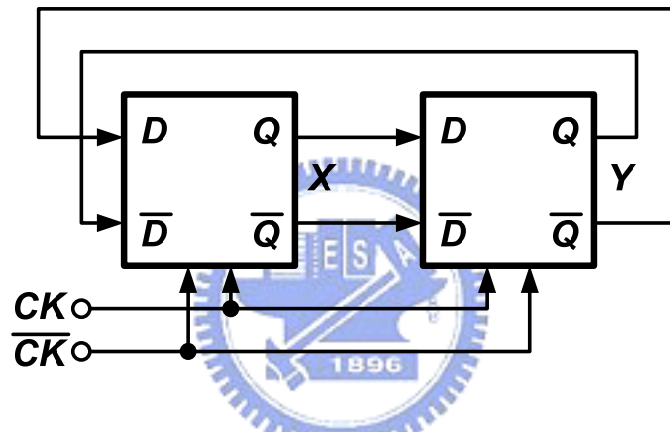
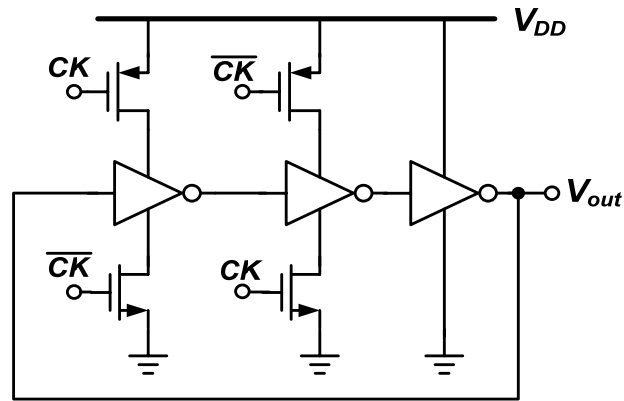
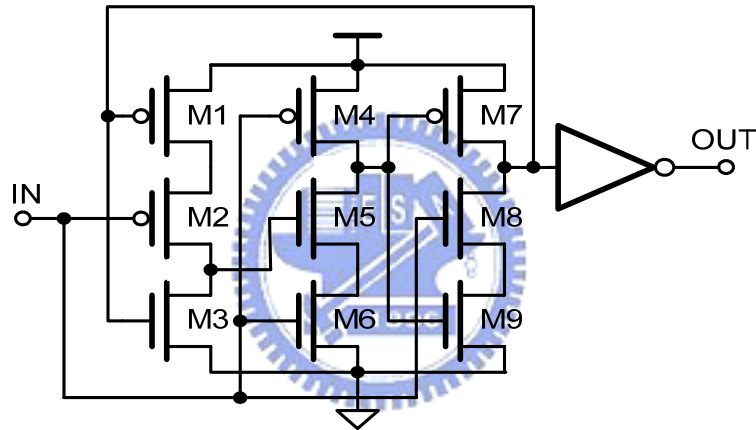


Fig.5-5 Divide-by-two circuit

We can also use dynamic latches in the high speed CMOS divide-by-two circuits. Fig. 5-6 shows two examples, in which the TSPC divider has been mentioned in the chapter 2. In the circuit of Fig. 5-6 (a), the first two CMOS inverters operate as dynamic latches controlled by CK and \overline{CK} and the third inverter provides the overall inversion required in the negative feedback loop. Fig. 5-6 (b) is a divide-by-two circuit realized by the TSPC register, which can achieve high speed. Lack of precise complementary or quadrature outputs is the disadvantage of both these circuits.



(a)



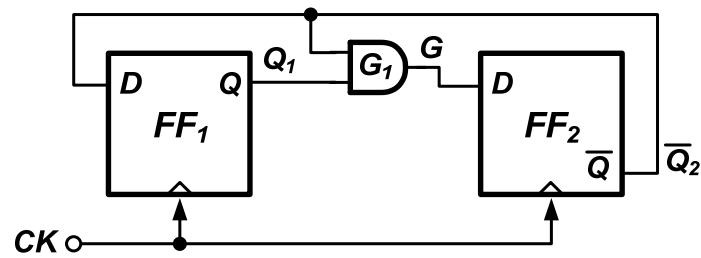
(b)

Fig.5-6 Dynamic dividers using (a) inverters, (b) TSPC

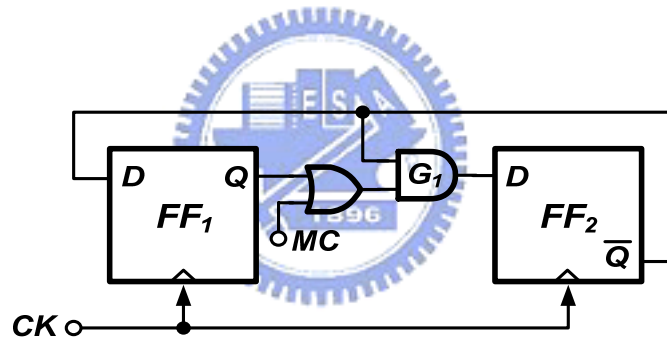
5.2.2 Dual-Modulus Dividers

Dual or multi-modulus dividers have been used in many phase-locked synthesizers. Such circuits divide the input frequency by one of the module according to a control input. A divide-by-2/3 circuit is a commonly used dual-modulus divider.

First, we consider a simple $\div 3$ circuit, shown in Fig. 5-7 (a). It uses two master-slave D-flipflops together with an AND gate to create three states: $Q_1\overline{Q_2} = 01, 10, 11$. Note that the state $Q_1\overline{Q_2} = 00$ can never occur (except at start-up).



(a)



(b)

Fig.5-7 (a) Divide-by-3 circuit (b) Divide-by-2/3 circuit

We can simply control Q_1 by interposing an OR gate between the first flipflop and the AND gate, so as to convert the topology of Fig. 5-7 (a) to a $\div 2/3$ circuit, shown in Fig. 5-7 (b). When MC is high, the divider is configured as a $\div 2$ circuit, and when MC is low, it is a $\div 3$ circuit. In addition, divide-by-three circuits are generally much slower than the divide-by-two counterparts. As shown in Fig. 5-7 (a), for example, following the clock edge on which $\overline{Q_2}$ must change, sufficient time

5.3 The Proposed ADPLL-Based Frequency Synthesizer

Based on the proposed ADPLL circuit, we can realize a frequency synthesizer for high speed clock generation by modifying the design of DCO counter in the PFD. As we know, the function of the counter in the PFD is just like the frequency divider in the feedback loop of the ADPLL. Hence, it can be implemented as an Integer- N synthesizer with multiple multiplication factors by using the adjustable counter length mechanism, as shown in Fig. 5-9.

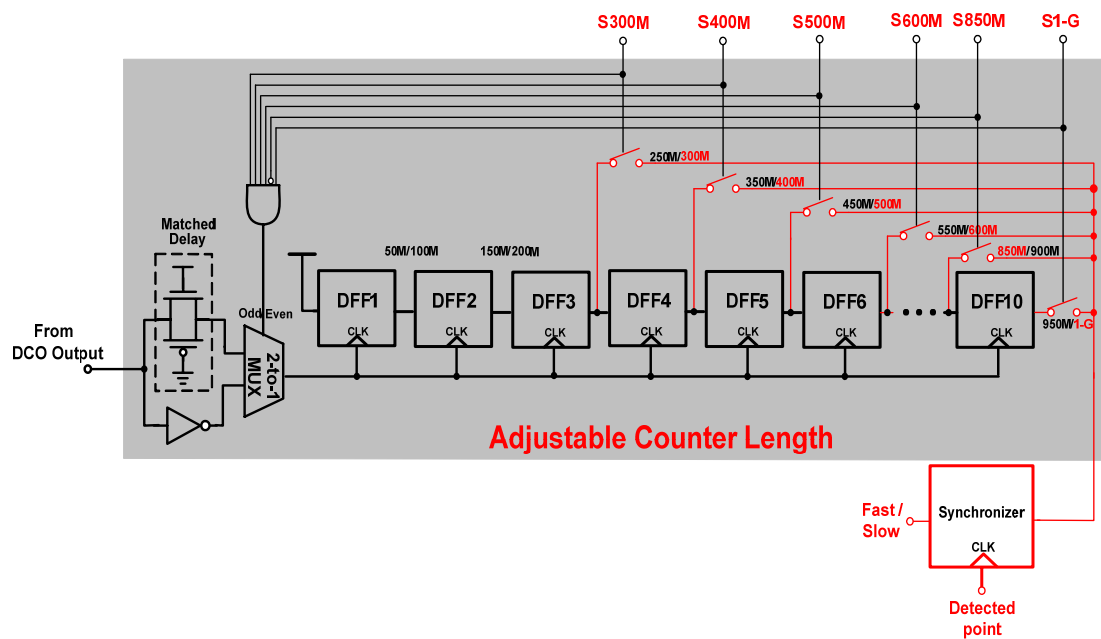


Fig.5-9 Adjustable DCO counter length

For example, if 500 MHz clock output is desired, the S500M input control signal will be asserted, then the fifth D Flip-Flop output signal would be selected to compare with the matched delay reference clock by the synchronizer made of a D flip-flop. Therefore, the ADPLL will generate 500 MHz clock output when the state is lock.

In addition, as mentioned above, since the middle point of the reference cycle is set as the detected point in our design, we also have to add an inverting path for DCO output so as to provide the output clocks with odd multiplications.

Besides, by referring to the different specifications in high speed DSP application, our frequency synthesizer has been designed for providing the 300M, 400M, 500M, 600M, 850M ,and 1GHz output clock. Finally, the relationship between input control signal and output clock frequency is summarized in Table 5-1.

Table 5-1 Frequency of output clock V.S. input control signal

<i>Input control signal</i>	<i>Frequency of the output clock</i>
S300M=1, others=0	300MHz
S400M=1, others=0	400MHz
S500M=1, others=0	500MHz
S600M=1, others=0	600MHz
S850M=1, others=0	850MHz
S1-G=1, others=0	1-GHz

5.4 Layout Implementation and Simulation Result

In this section, we will show the layout and simulation result of our ADPLL design. In the layout implementation phase, we should take care about the floor plan first, then, also consider the shape of each block, as well as the matching and connection among them.

Fig. 5-10 and Fig. 5-11 show the layout implementation and floor plan respectively. The main signal path among these building blocks in ADPLL is roughly counterclockwise, in this way we can place these blocks compactly. Fig. 5-12 shows the area occupation ratio of each building block in ADPLL; 32% of the overall area is for DCO and its replica delay element, 56% for control unit, and 12% for PFD. With regard to power-consumption distribution in the ADPLL, as shown in Fig. 5-13, because of the high operating frequency of DCO, the DCO and its replica circuit dissipate a large part of total power dissipation in spite of their smaller area occupation than all of other circuits in the ADPLL.

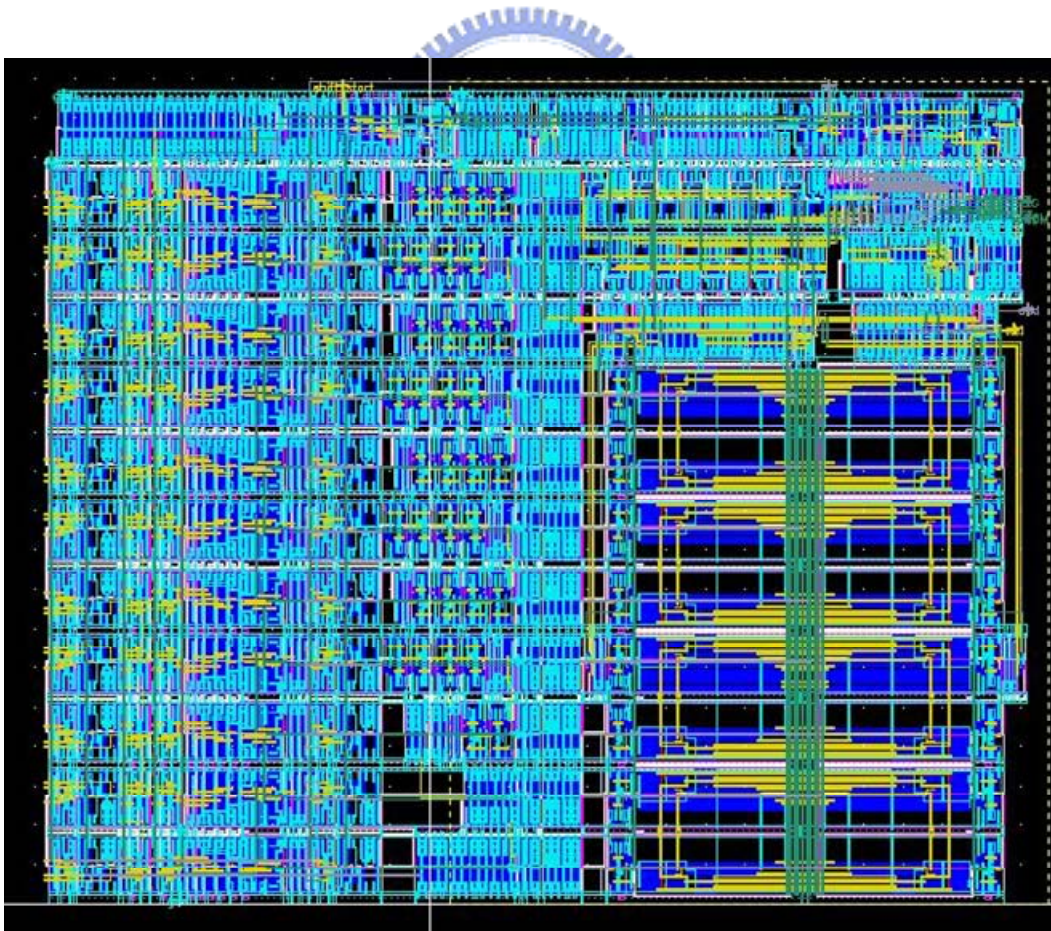


Fig.5-10 Layout of the ADPLL

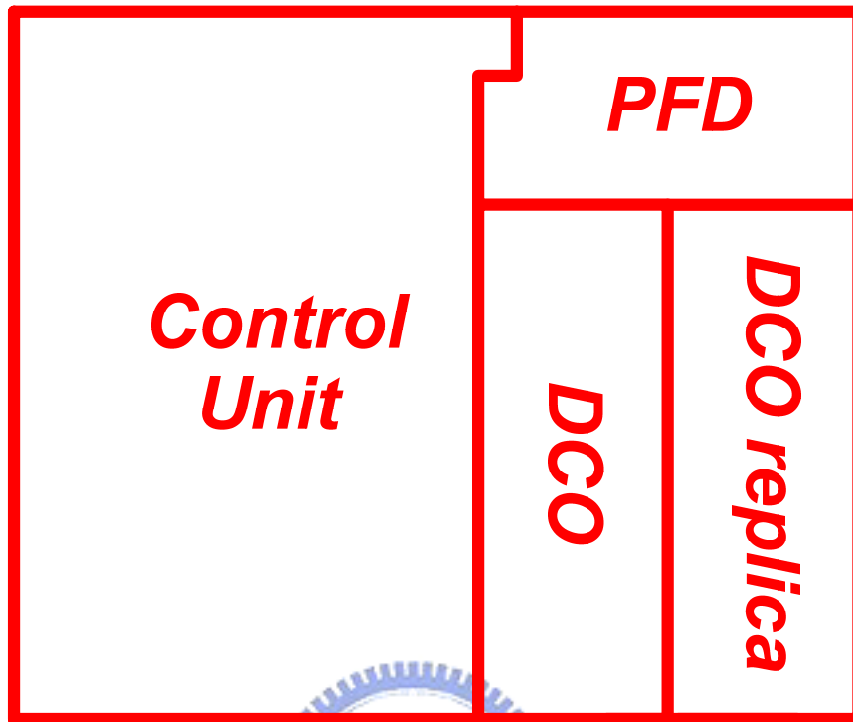


Fig.5-11 Floor Plan of the ADPLL

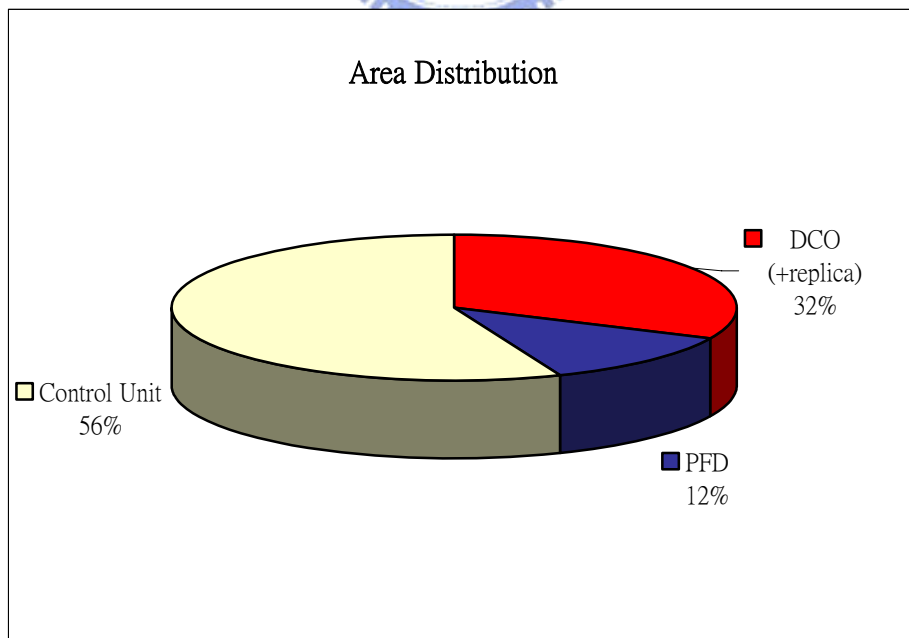


Fig.5-12 Area Distribution of the ADPLL

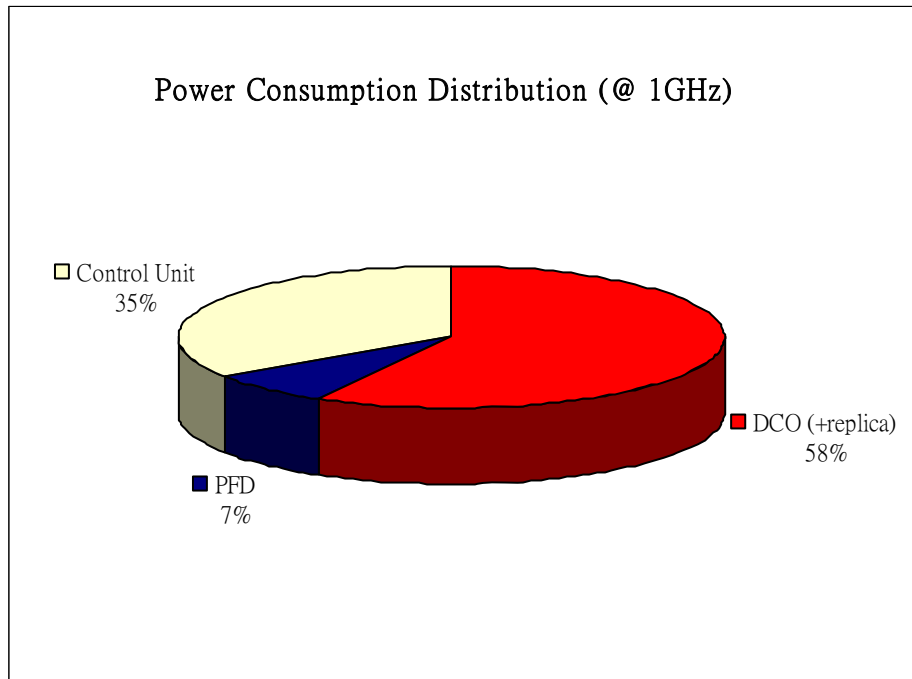


Fig.5-13 Power-Consumption Distribution of the ADPLL

Thereupon, we will present the overall simulation results with different operating frequency, and the input reference clock frequency is always set as 50 MHz. Fig. 5-14, Fig. 5-15, Fig. 5-16, Fig. 5-17, Fig. 5-18, and Fig. 5-19 show the lock process of ADPLL in output clock of 300 M, 400 M, 500 M, 600 M, 850 M, and 1-GHz respectively. Each figure includes three sub-figures, they are the overall lock process, the zoom in of lock process, and the locked state of ADPLL individually.

Besides, within each sub-figure, there are four main signals are shown: the reference clock, DCO output clock, LOCK signal, and value of DCO control word. The reference clock is from the matched delay clock of the input reference signal.

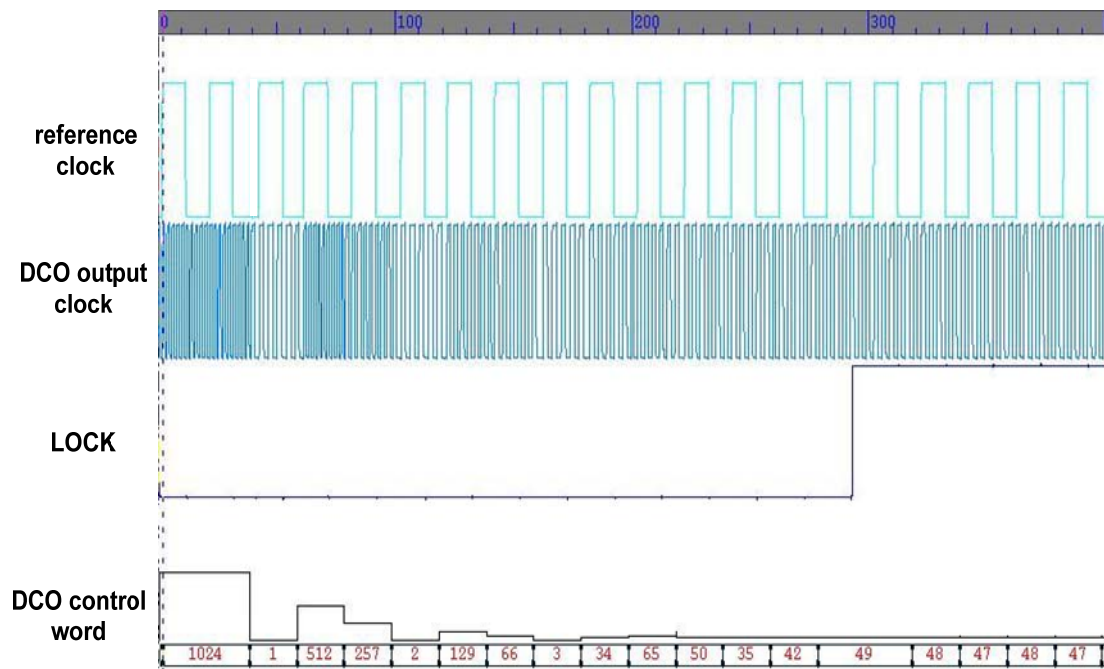


Fig.5-14 (a) The overall lock process of 300MHz target frequency.

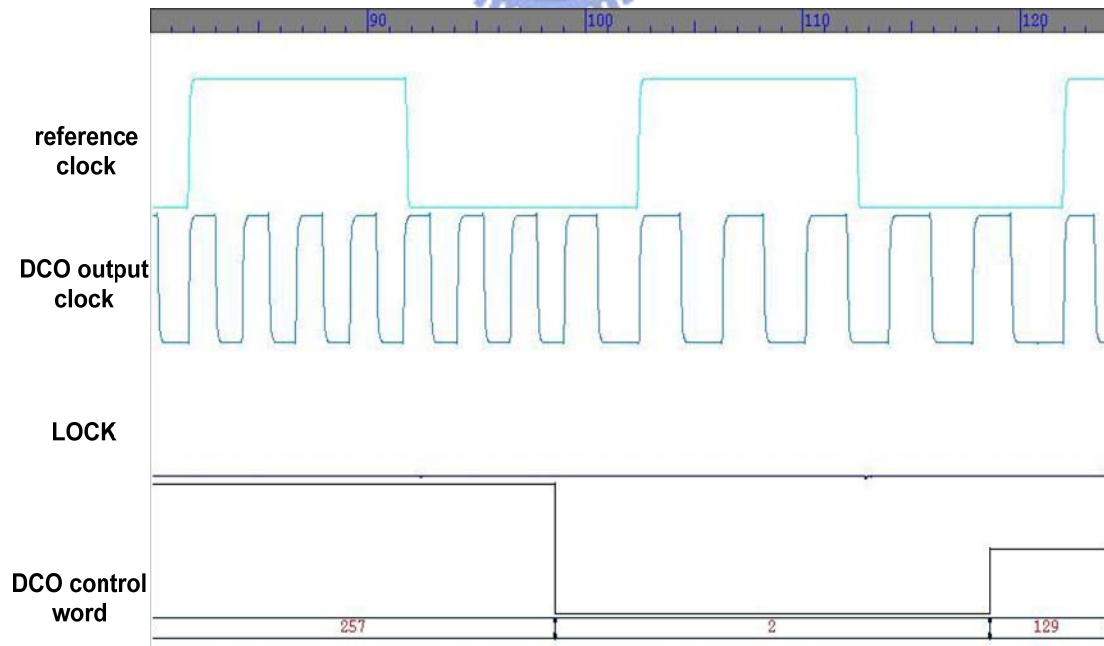


Fig.5-14 (b) The zoom in of lock process of 300MHz target frequency.

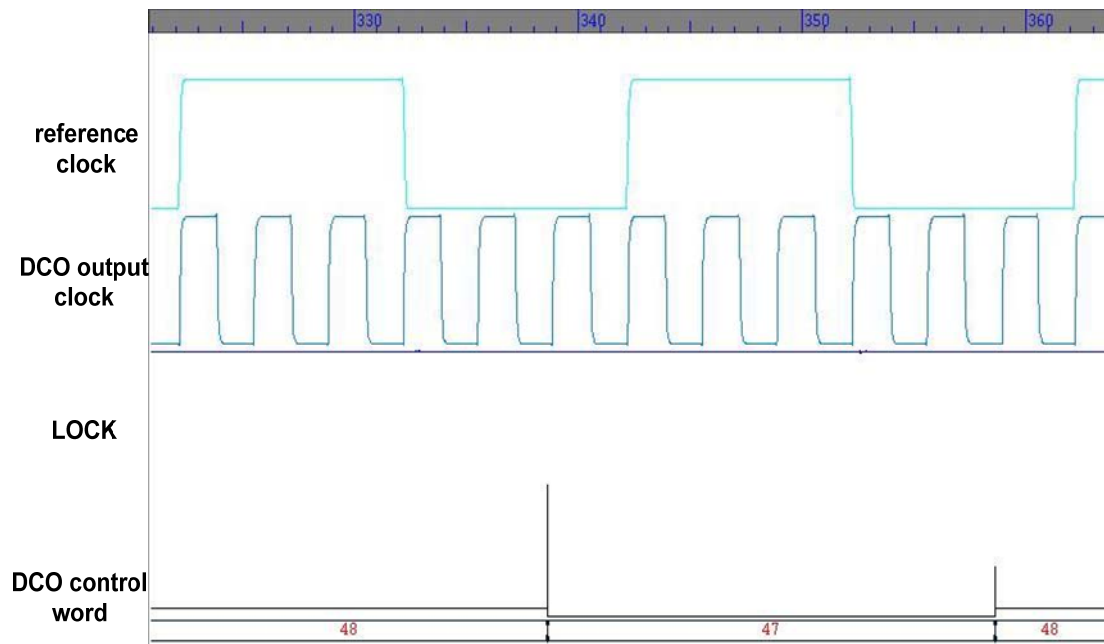


Fig.5-14 (c) The locked state of 300MHz target frequency.

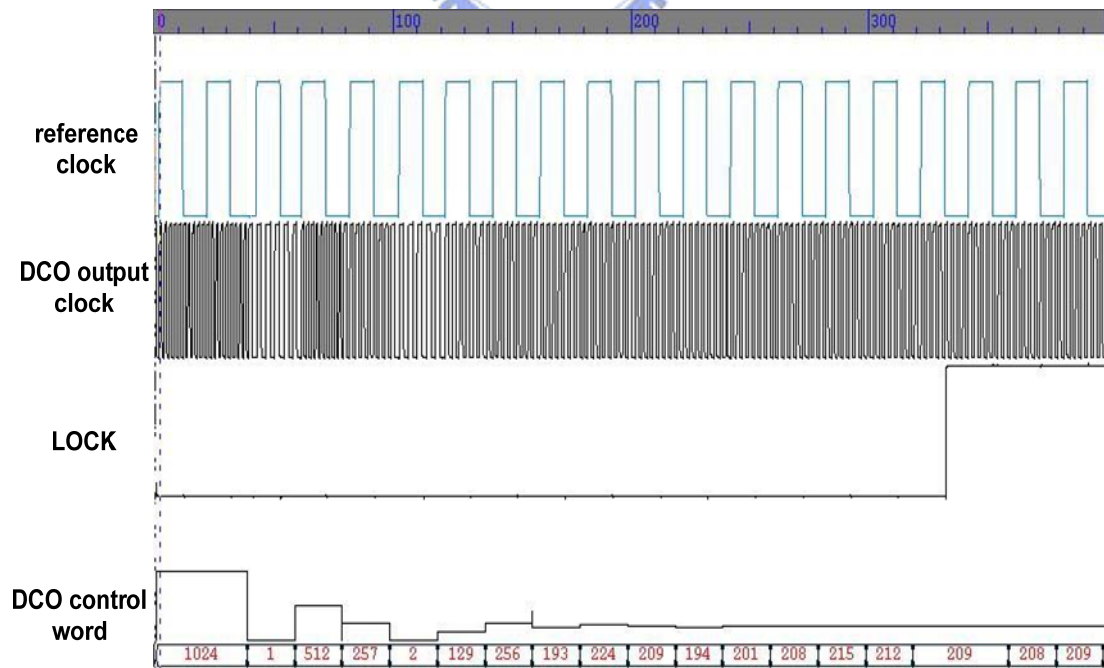


Fig.5-15 (a) The overall lock process of 400MHz target frequency.

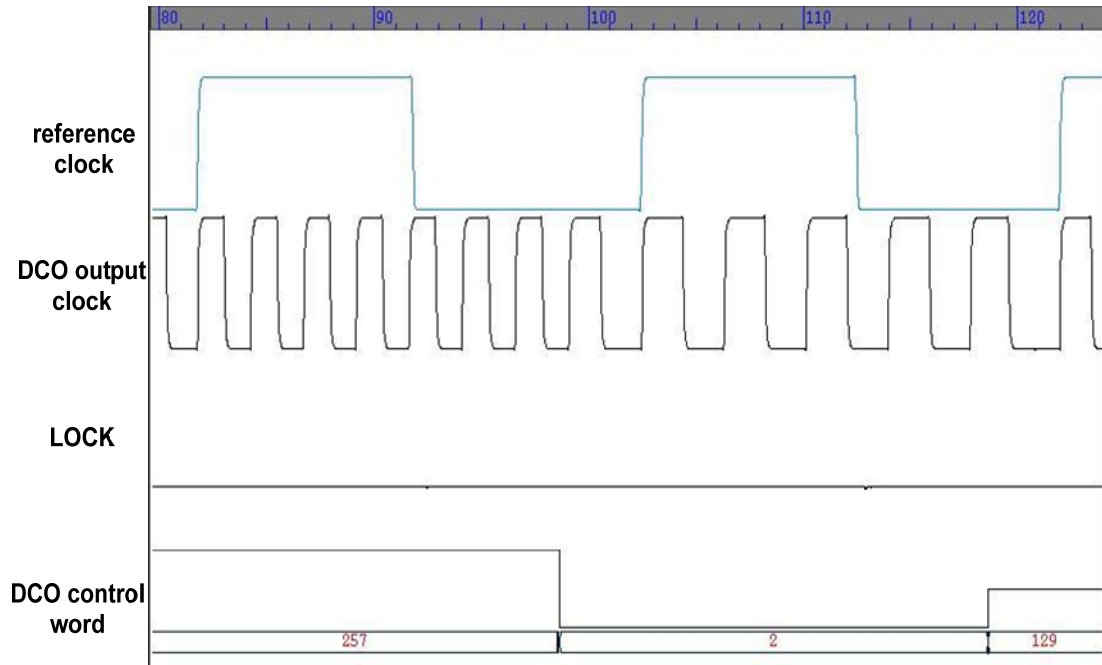


Fig.5-15 (b) The zoom in of lock process of 400MHz target frequency.

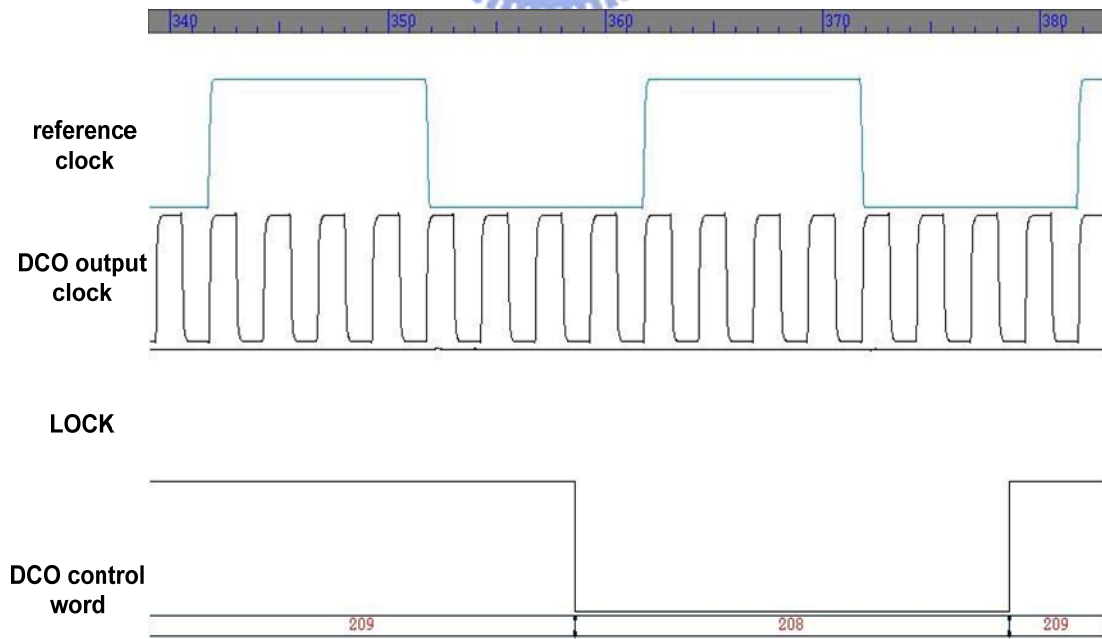


Fig.5-15 (c) The locked state of 400MHz target frequency.

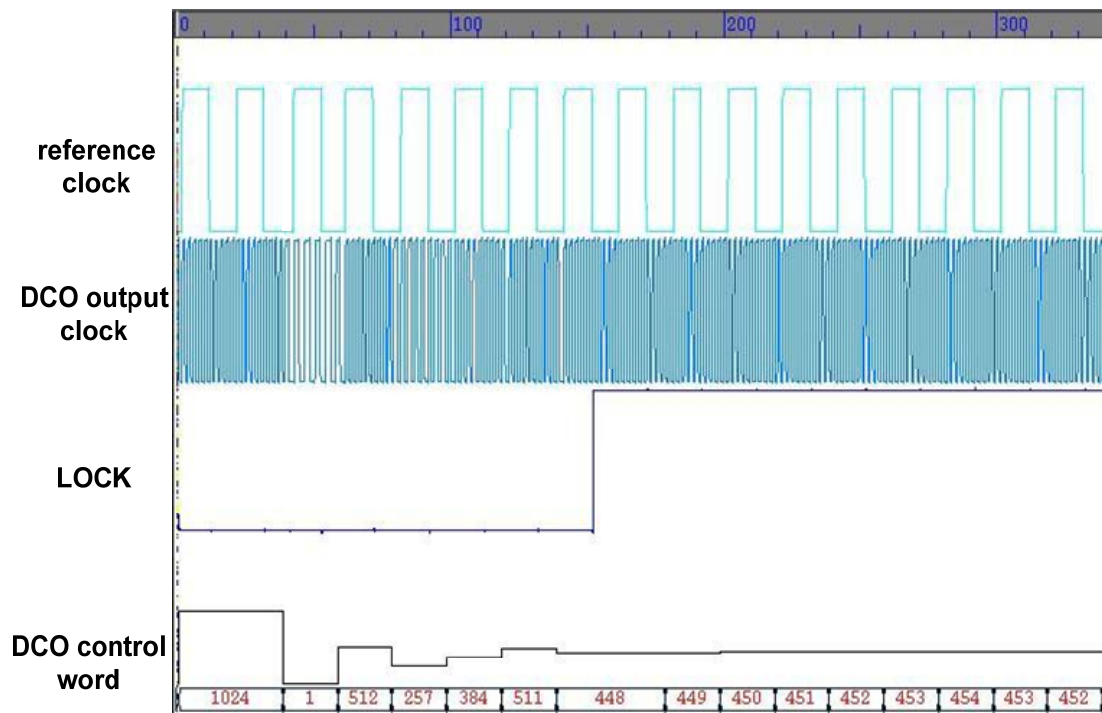


Fig.5-16 (a) The overall lock process of 500MHz target frequency.

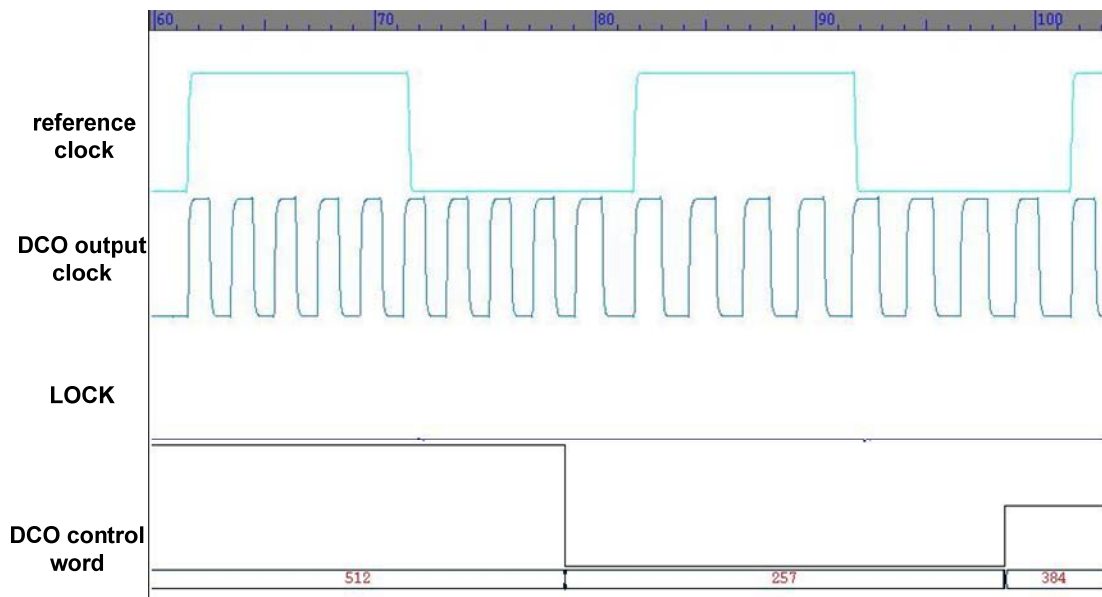


Fig.5-16 (b) The zoom in of lock process of 500MHz target frequency.

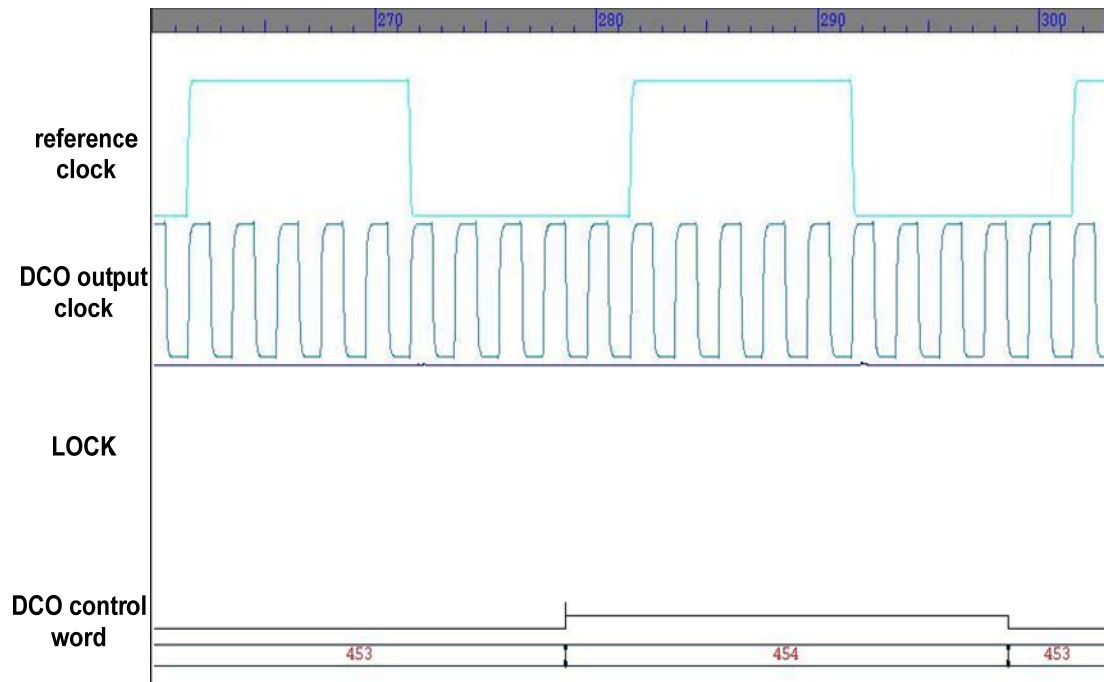


Fig.5-16 (c) The locked state of 500MHz target frequency.

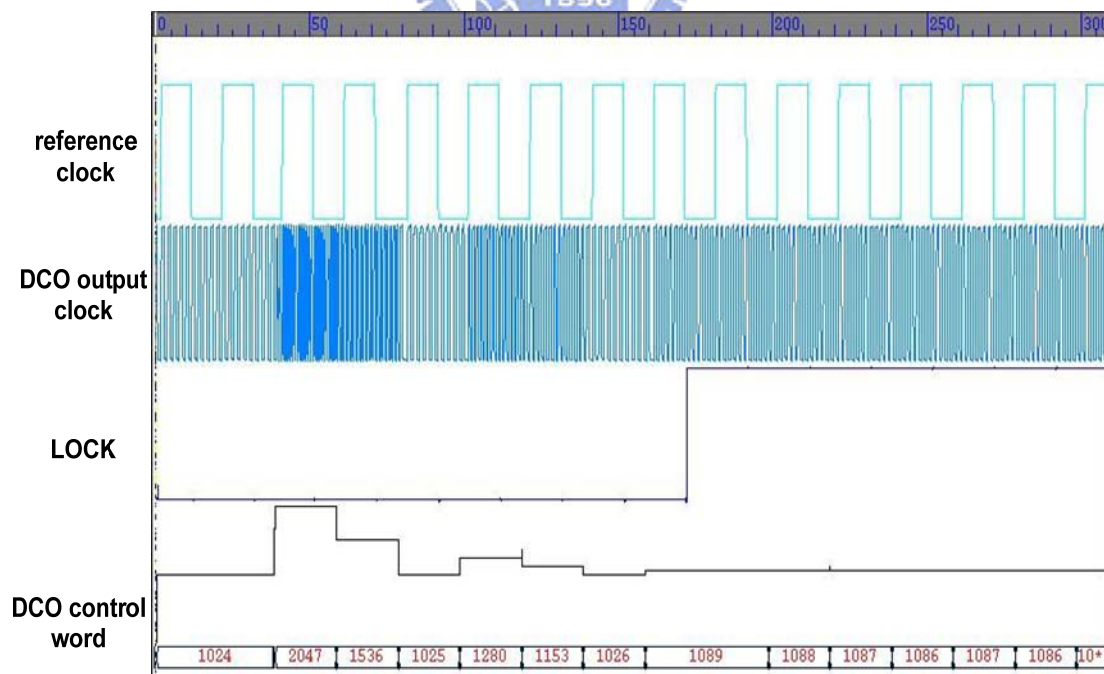


Fig.5-17 (a) The overall lock process of 600MHz target frequency.

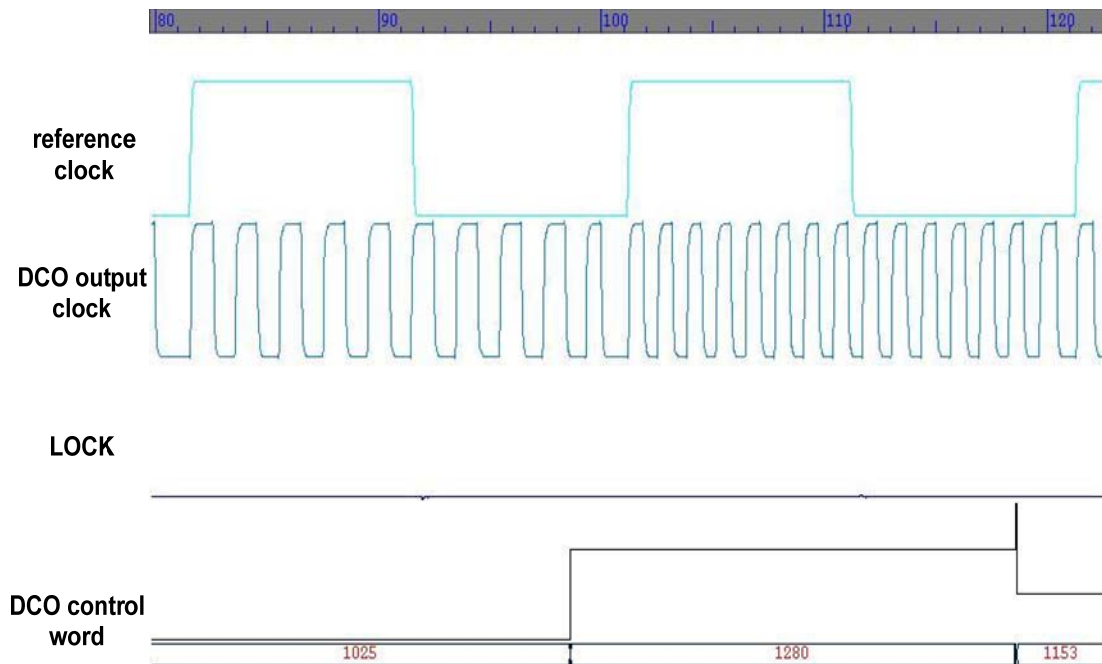


Fig.5-17 (b) The zoom in of lock process of 600MHz target frequency.

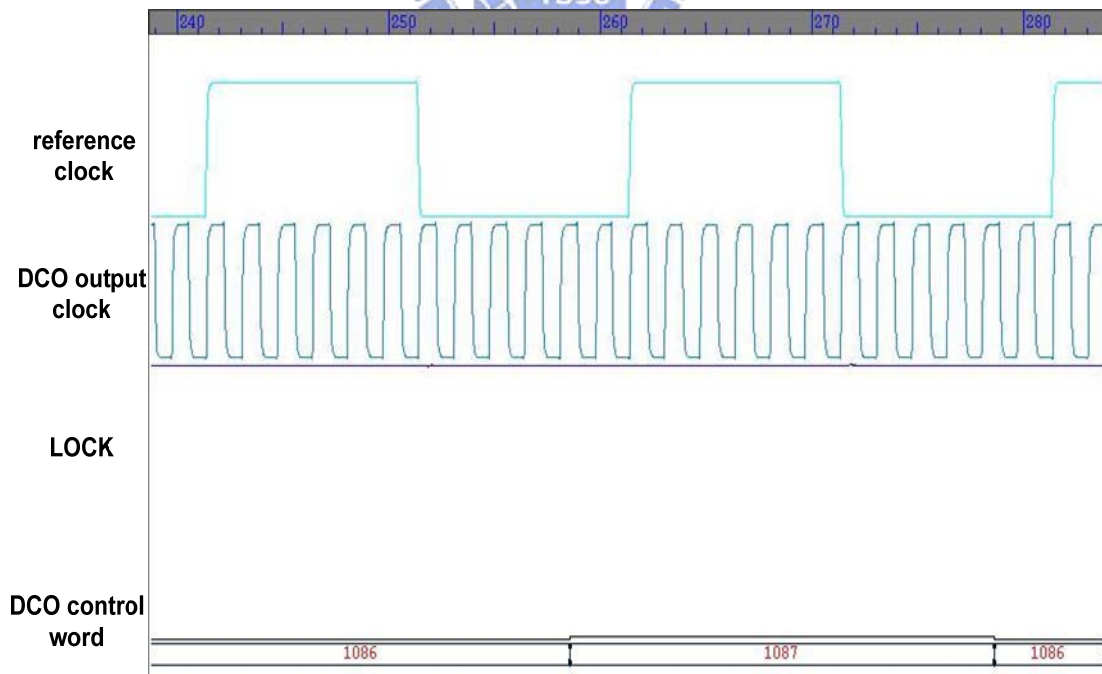


Fig.5-17 (c) The locked state of 600MHz target frequency.

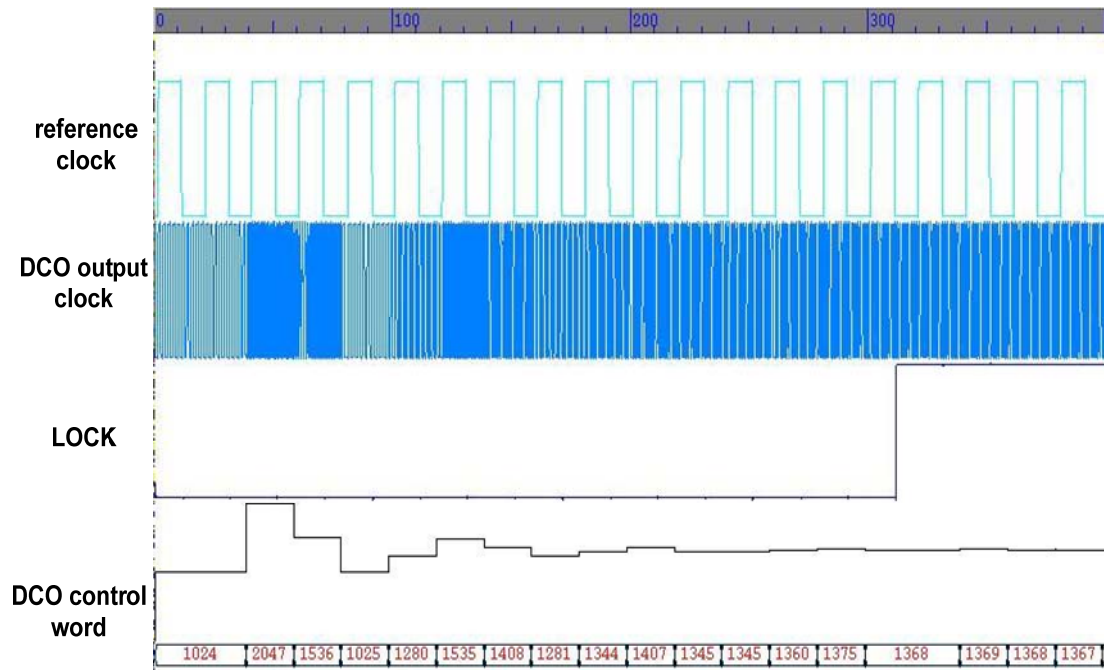


Fig.5-18 (a) The overall lock process of 850MHz target frequency.

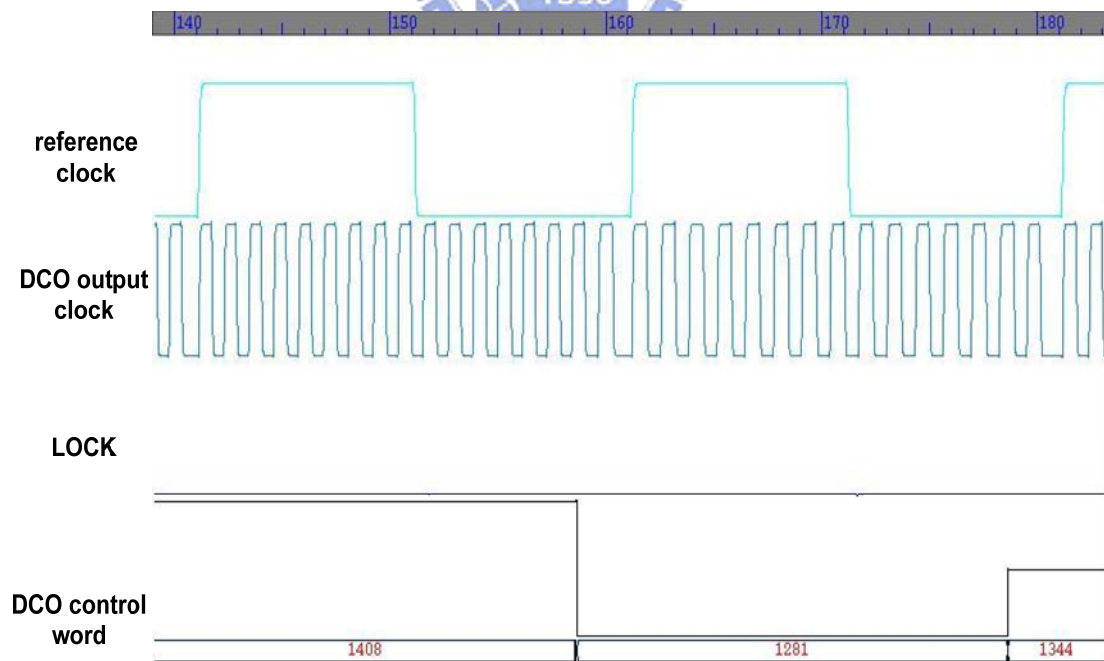


Fig.5-18 (b) The zoom in of lock process of 850MHz target frequency.

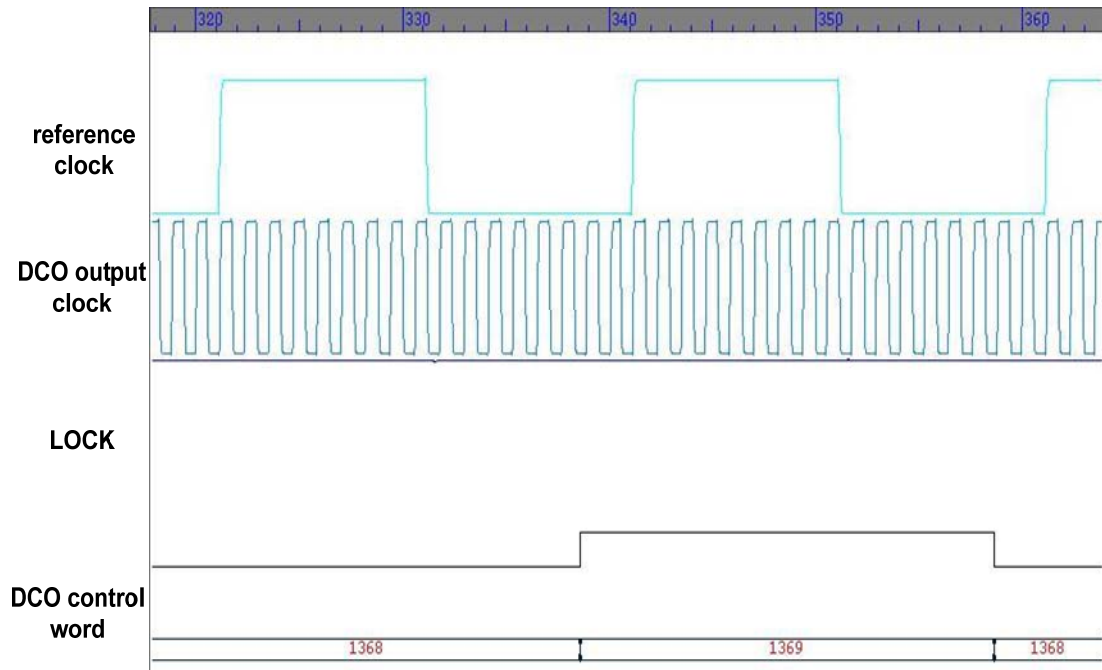


Fig.5-18 (c) The locked state of 850MHz target frequency.

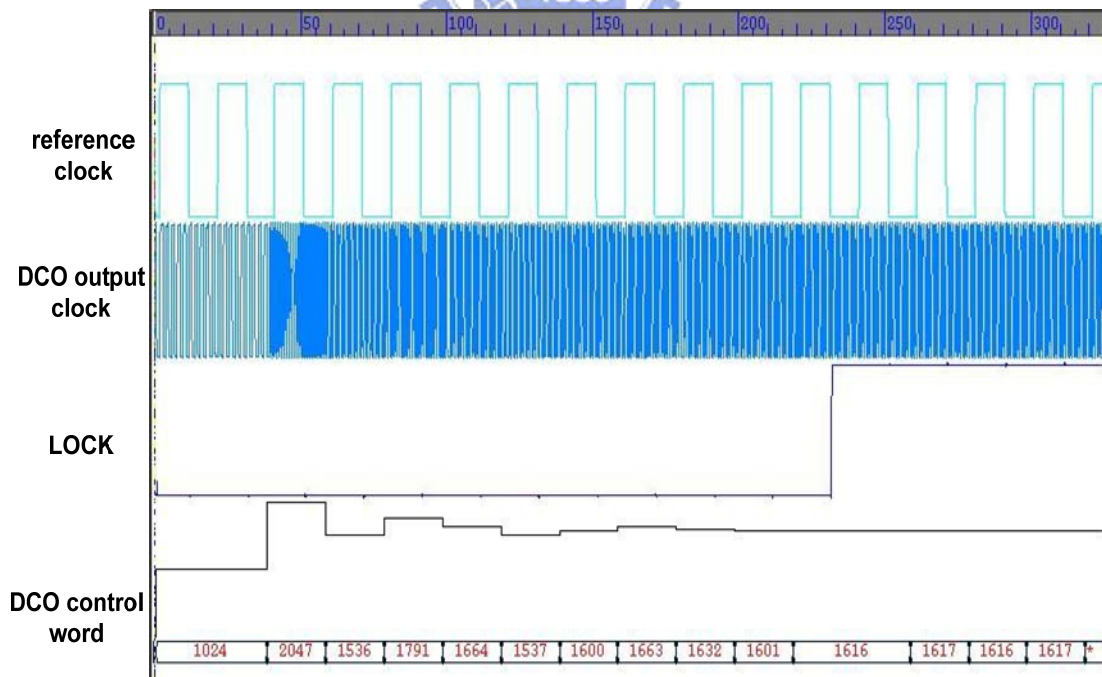


Fig.5-19 (a) The overall lock process of 1GHz target frequency.

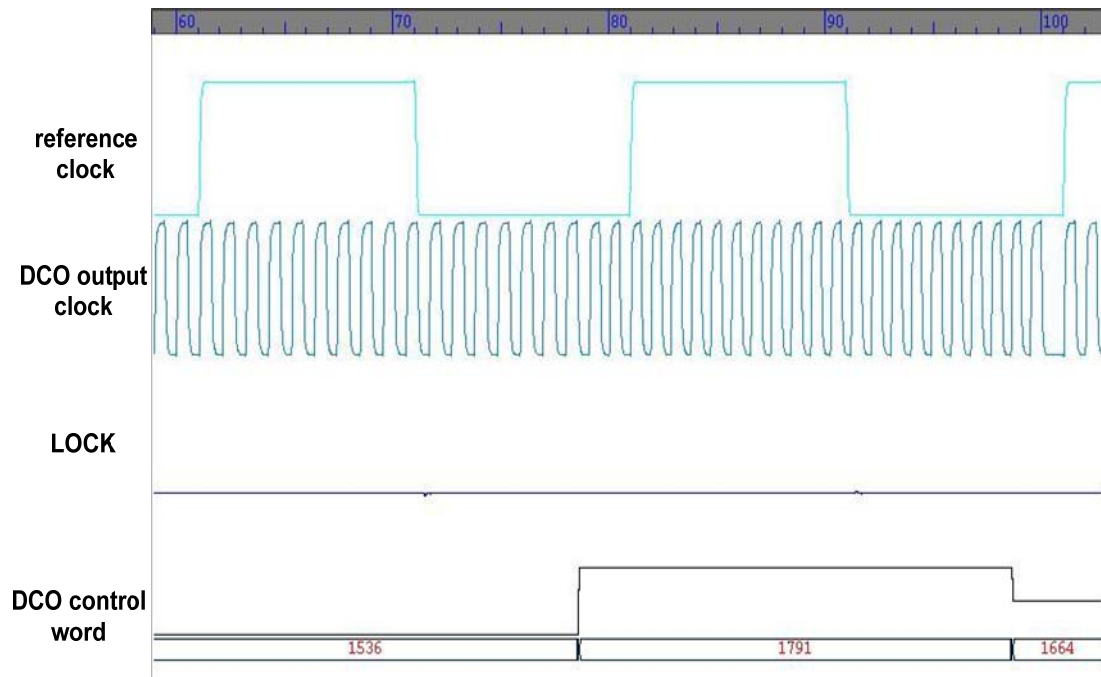


Fig.5-19 (b) The zoom in of lock process of 1GHz target frequency.

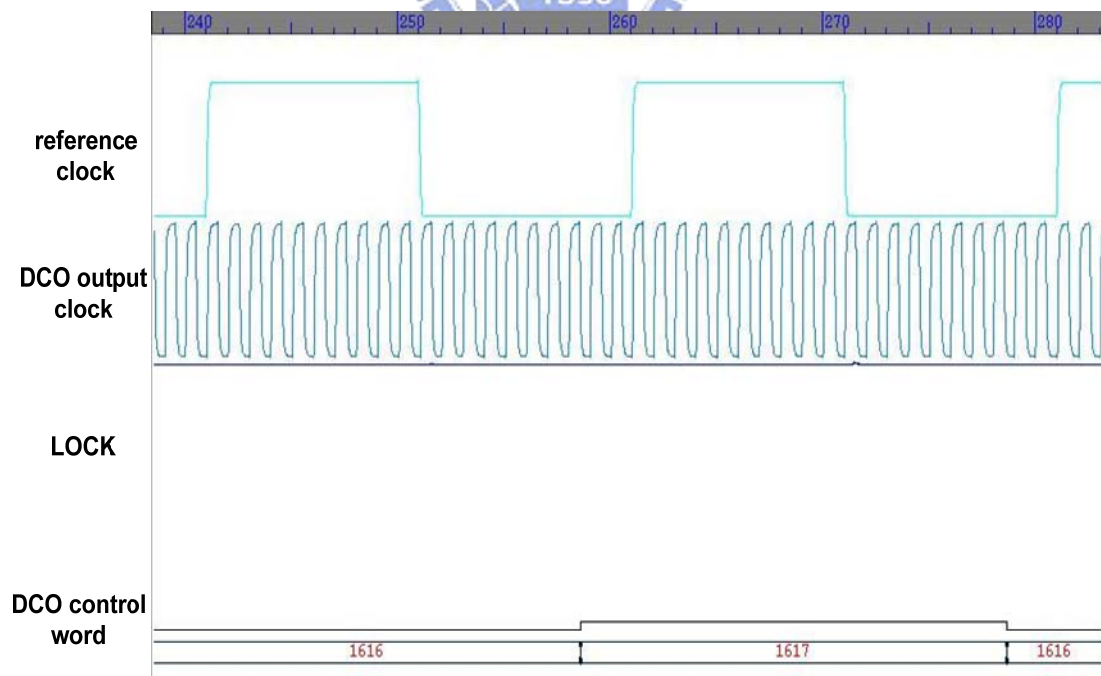


Fig.5-19 (c) The locked state of 1GHz target frequency.

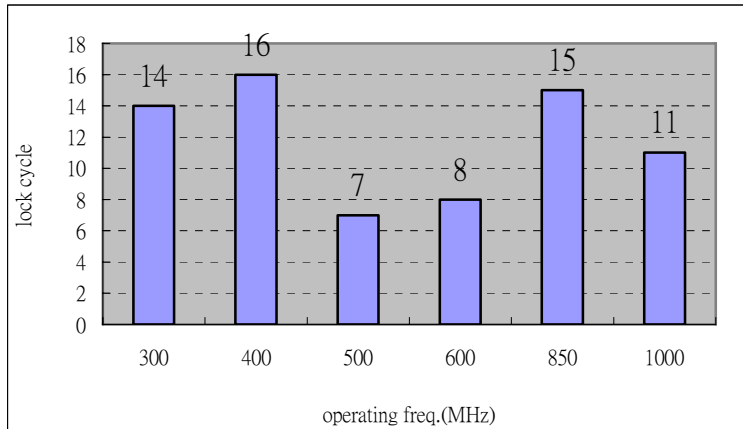


Fig.5-20 Lock cycle VS. operating frequency

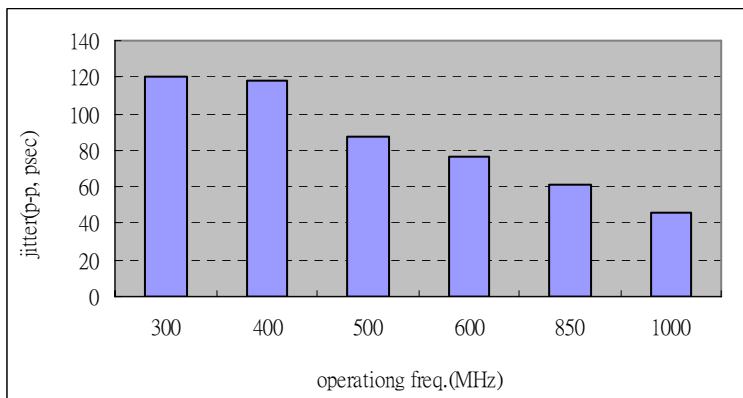


Fig.5-21 Jitter VS. operating frequency

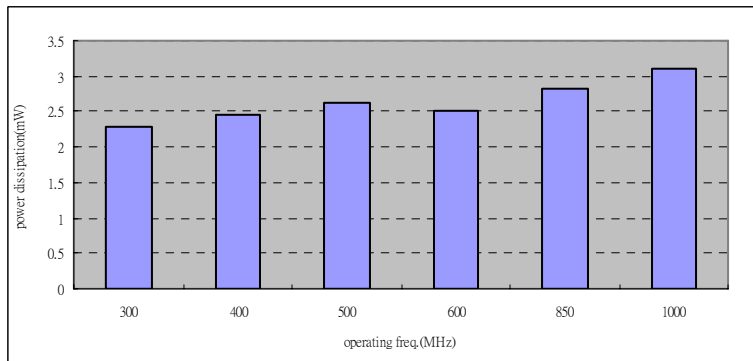


Fig.5-22 Power dissipation VS. operating frequency

Finally, Fig. 5-20, Fig. 5-21, and Fig. 5-22 present performance summaries of the ADPLL-based frequency synthesizer in different operating frequency. As shown in Table 5-2, compare to other ADPLL circuit design, our ADPLL owns the largest frequency range and the monotonicity characteristic due to the new DCO design. In addition, our ADPLL needs the smallest area and power cost because of its simplicity, and also has the fewer lock cycles by using the modified PFD.

Table 5-2 Performance comparison of ADPLL

<i>Design</i>	<i>JSSC04 [3]</i>	<i>JSSC03 [23]</i>	<i>JSSC03 [28]</i>	<i>This Work</i>
<i>Min. freq.(MHz)</i>	152	45	30	260
<i>Max. freq.(MHz)</i>	366	510	650	1150
<i>Power Consumption</i>	24mW @ 366MHz	100mW @ 500MHz	7mW @ 240MHz	3.1mW @ 1GHz
<i>Area</i>	0.07mm ²	0.71mm ²	0.182mm ²	0.012 mm ²
<i>Lock Cycle</i>	N.A.	<46	N.A.	<16
<i>Jitter(p-p)</i>	0.775~1.2ns @ 320/64 MHz	70ps @ 450MHz	71ps @ 240MHz	46ps @ 1GHz
<i>Process</i>	0.35	0.35	0.13	0.13

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, an ADPLL-based frequency synthesizer with low power and area cost, wide frequency range, and short lock cycle is proposed.

A DCO with features of wide frequency range, and low power consumption is proposed. By using the new type digitally controlled delay element (DCDE), a digitally controlled oscillator (DCO) with characteristics of its monotonicity is presented, which makes the DCO design more straightforward. In addition, compared with the conventional architecture, a dual-mode one-cycle PFD used to reduce the lock cycle time is also presented.

Table 6-1. Performance Summary of the Frequency Synthesizer

<i>ADPLL-based Frequency Synthesizer</i>	
<i>Reference clock</i>	50MHz
<i>Output Clock Frequency</i>	300MHz / 400MHz / 500MHz / 600MHz / 850MHz / 1GHz
<i>Jitter (p-p)</i>	46ps @ 1GHz 120ps @ 300MHz (worst case)
<i>Locked time</i>	≤16 input cycles

<i>Power consumption</i>	3.1mW @ 1GHz
<i>Area</i>	100um x 120um
<i>Supply voltage</i>	1.2V
<i>Process</i>	TSMC 0.13um CMOS process
DCO	
<i>Frequency Range</i>	260MHz ~ 1.15GHz
<i>Resolution</i>	11bits (0.4ps ~ 20ps / LSB)
<i>Power consumption</i>	0.9mW @ 1GHz

In conclusion, with the specification, the proposed ADPLL-based frequency synthesizer is suitable for high speed clock generation in high speed DSPs applications.



6.2 Future Work

In the recent year, low power is a more and more important issue in circuit design. However, most techniques to reduce power dissipation of integrated circuits are to choose system and circuit parameters at design time. In fact, in some applications, it is a more efficiency way of adjusting the circuit during operation, such as: voltage scaling or lowering the operating clock frequency. The related control schemes have been proposed in [24]-[27].

To reduce the power dissipation, it usually needs different voltage or frequency in a low power system. Hence, in our future work, we can take advantage of the adaptive voltage scaling (AVS) scheme in our ADPLL to decrease the power consumption further, as shown in Fig. 6-1. The adaptive voltage scaling circuit can

provide several different supply voltage for the DCO, which consumes a great part of the total power dissipation in the ADPLL. Therefore, by scaling down the supply voltage of DCO, we can save much power dissipation while still meeting the same specification.

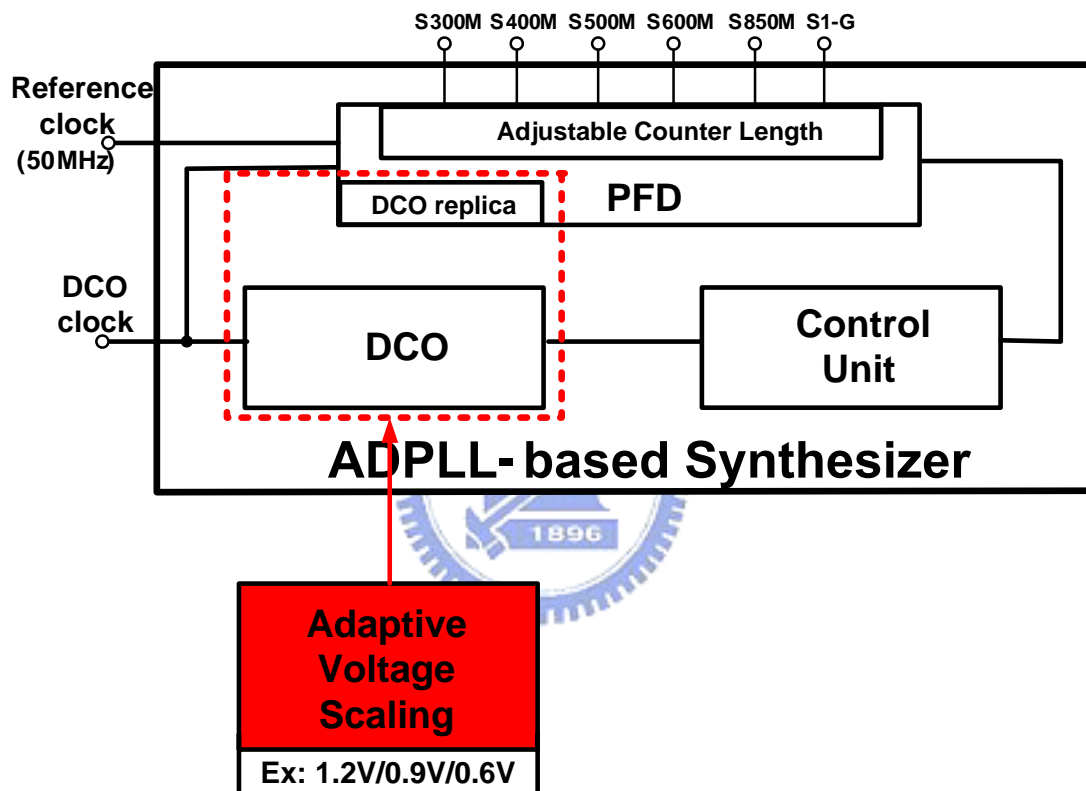


Fig.6-1 The ADPLL with AVS

Reference

- [1] Behzad Razavi, Design of Analog CMOS Integrated Circuits, Boston: McGraw-Hill, 2001.
- [2] T. Olsson, and P. Nilsson, "A digital PLL made from standard cell," in *Proc. 15th ECCTD*, pp. 277-280, 2001.
- [3] T. Olsson, and P. Nilsson, "A digitally controlled PLL for SoC applications," *IEEE J. Solid-State Circuits*, Vol. 39, pp. 751 - 760, May 2004.
- [4] R. E. Best, *Phase-Locked Loops: Theory, Design and Applications*, 2nd ed. New York: McGraw-Hill, 1993.
- [5] 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, Apr. 1995.
- [6] J.-S. Chiang, and K.-Y. Chen, "A 3.3 V all digital phase-locked loop with small DCO hardware and fast phase lock," in *ISCAS*, vol. 3, pp. 554-557, 1998.
- [7] Eric Pascal Roth, "All-Digital Standard-Cell Based Audio Clock Synthesis," *Ph.D Dissertation of ETH, No. 15667*, E.E. Department, Zurich, Swiss, 2004
- [8] F. M. Gardner, "Phase-lock Techniques," 2nd Edition. John Wiley & Sons, New York, NY 1979.
- [9] J. Craninckx, M. Steyaert, "Wireless CMOS Frequency Synthesizer Design," Kluwer Academic Publishers, 1998.
- [10] D. C. Cox and R. P. Leek, "Component signal separation and recombination for linear amplification with nonlinear components," *IEEE Trans. Comm.*, vol. 23, pp.1281-1287, Nov. 1975.

- [11] J. Craninckx, M. Steyaert, "Low-noise voltage-controlled oscillators using enhanced LC-tanks," *IEEE Trans. On Circuit and System II*, vol. 42, no. 12, Dec. 1995.
- [12] Paul Gray, Iasson F. Vassiliou, "Top-Down Design of a Phase Locked Loop For a Video Driver System," University of California at Berkeley master thesis.
- [13] M. Saint-Laurent and M. Swaminathan, "A digitally adjustable resistor for path delay characterization in high frequency microprocessors," in *Proc. Southwest Symp. Mixed-Signal Design*, 2001, pp. 61–64.
- [14] M. Maymandi-Nejad and M. Sachdev, "A digitally programmable delay element: Design and analysis," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 11, no. 5, pp. 871–878, Oct. 2003.
- [15] M. Maymandi-Nejad, and M. Sachdev, "A monotonic digitally controlled delay element" *IEEE J. Solid-State Circuits*, Vol. 40, no. 11, pp. 2212-2219, Nov. 2005.
- [16] Behzad Razavi, RF Microelectronics, NJ: Prentice Hall, 1998.
- [17] R. G. Meyer, R. Eschenbach, and W. M. Edgerley, "A Wideband Feedforward Amplifier," *IEEE J. Solid-State Circuits*, Vol. 9, pp. 422-488, Dec. 1974.
- [18] E. E. Eid, F. M. Ghannouchi, and F. Beaugard, "Optimal Feedforward Linearization System Design," *Microwave J.*, pp. 78-86, November 1995.
- [19] Amr M. Fahim, "A Compact, Low-Jitter Digital PLL," in *Proc. IEEE ESSCIRC*, pp. 101-104, Sep. 2003.
- [20] T.-Y Hsu and C.-Y Lee, "An All-Digital Phase-Locked Loop (ADPLL)-Based Clock Recovery Circuit," *IEEE J. Solid-State Circuits*, Vol. 34, no. 8, pp. 1063-1073, Aug. 1999.

- [21] T.-Y Hsu C.-C Wang, and C.-Y Lee, "Design and Analysis of a Portable High-Speed Clock Generator," *IEEE Trans. Circuit and Syst. II, Analog and Digital Signal Processing*, Vol. 36, no. 10, pp. 1574-1581, Oct. 2001.
- [22] Pao-Lung Chen, "The Study of Portable Digitally Controlled Oscillator and Dynamic Frequency Counting Loop," *Ph.D Dissertation of Institute of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, R. O. C.*, Nov. 2005.
- [23] C.-C Chung and C.-Y Lee, "An All-Digital Phase-Locked Loop for High-speed Clock Generation," *IEEE J. Solid-State Circuits*, Vol. 38, no. 2, pp. 347-351, Feb. 2003.
- [24] P. Macken et al., "A voltage reduction technique for digital systems," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 1990, pp.238–239.
- [25] S. Sakiyama et al., "A lean power management technique: The lowest power consumption for the given operating speed of LSIs," in *Symp. VLSI Circuits Dig. Tech. Papers*, June 1997, pp. 99–100.
- [26] K. Suzuki et al., "A 300 MIPS/W RISC core processor with variable supply-voltage scheme in variable threshold-voltage CMOS," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 1997, pp.587–590.
- [27] T. Burd et al., "A dynamic voltage scaled microprocessor system," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2000, pp. 294–295.
- [28] J. G. Maneatis, J. Kim, I. McClatchie, J. Maxey and M. Shankarards, "Self-Biased High-Bandwidth Low-Jitter 1-to-4096 Multiplier Clock Generator PLL," *IEEE J. Solid-State Circuits*, Vol. 38, no. 11, pp. 1795-1803, Nov. 2003.

Vita

PERSONAL INFORMATION

Birth Date: January. 04, 1981

Birth Place: Taipei, Taiwan, R.O.C.

Address: Department of Electronics Engineering
National Chiao Tung University
1001 Ta-Hsueh Road
Hsin-chu, Taiwan 30010, R.O.C.

E-Mail Address: ujcc.ee92g@nctu.edu.tw

EDUCATION

B.S. [2003] Department of Electronical Engineering, National Cheng-Kung
University.

M.A. [2006] Institute of Electronics, National Chiao-Tung University.

