## 國立中正大學

資訊工程研究所碩士論文

可支援動態電壓與頻率調整之快速 鎖定 0.5V/1.0V 全數位鎖相迴路設計

A 0.5V/1.0V Fast Lock-in ADPLL for Supporting Dynamic Voltage and Frequency Scaling

研究生: 蘇煒翔

指導教授: 鍾菁哲 博士

中華民國 一零二 年 八 月

國立中正大學碩士班研究生

學位考試同意書

本人所指導 資訊工程學系

研究生 蘇煒翔 所提之論文

可支援動態電壓與頻率調整之快速鎖定 0.5V/1.0V 全數位鎖相迴 路設計 (A 0.5V/1.0V Fast Lock-in ADPLL for supporting dynamic voltage and frequency scaling)

同意其提付 碩 士學位論文考試

指導教授 董重著 折 簽章 102年5月2月日

國立中正大學碩士學位論文考試審定書

#### 資訊工程學系

#### 研究生蘇煒翔所提之論文

<u>可支援動態電壓與頻率調整之快速鎖定 0.5V/1.0V 全</u> <u>數位鎖相迴路設計 (A 0.5V/1.0V Fast Lock-in ADPLL</u> for supporting dynamic voltage and frequency <u>scaling</u>) 經本委員會審查,符合碩士學位論文標準。

學位考試委員會
7 集 人\_\_\_\_ 香顺裕 簽章 委 員 簽章 指 教 月 15 中華民國

日

#### 博碩士論文授權書

(本聯請裝訂於論文紙本書名頁前空白處,供學校圖書館做為授權管理用) ID:101CCU00392079

本授權書所授權之論文為授權人在國立中正大學(學院)資訊工程研究所系所 \_\_\_\_\_ 組 101 學年度第 二學期取得 項士學位之論文。

論文題目: <u>可支援動態電壓與頻率調整之快速鎖定0.5V/1.0V全數位鎖相迴路設計</u>

指導教授: <u>鍾菁哲</u>, Ching-Che Chung

茲同意將授權人擁有著作權之上列論文全文(含摘要),提供讀者基於個人非營利性質之線上檢 索、閱覽、下載或列印,此項授權係非專屬、無償授權國家圖書館及本人畢業學校之圖書館, 不限地域、時間與次數,以微縮、光碟或數位化方式將上列論文進行重製,並同意公開傳輸數 位檔案。

紙本論文:茲同意將授權人擁有著作權之上列論文全文(含摘要),提供讀者基於個人非營利性 質之閱覽或列印,此項授權系非專屬、無償授權國立中正大學圖書館做為編目上架及公開陳列 閱覽使用。

□ 校内外立即開放

□ 校內立即開放,校外於 2018 年 08 月 12 日後開放

✓ 校內於 2018 年 08 月 12 日;校外於 2018 年 08 月 12 日後開放
 □ 其他

授權人:蘇煒翔

in in

簽名: 新樟和

日期: /02年 8月 (2日

#### 摘要

近年來,隨著智慧型手機、無線感測網路等行動裝置的發展,以及使用生 理訊號紀錄晶片的生醫電子應用逐漸地被開發,由電池供電(Battery-Powered) 的系統已經是越來越普遍。

動態電壓與工作頻率調整(Dynamic Voltage and Frequency Scaling, DVFS) 技術,為目前已經被廣泛應用在 SoC 的電源管理上,以降低動態功率消耗的 成熟技術。當電路工作電壓(Supply Voltage)降低時,其所消耗之動態功率消耗 也隨之降低,因此降低工作電壓為降低系統功率消耗最有效率的手段。

然而工作電壓降低,也伴隨著電路速度變慢與易受到製程飄移、電壓飄移 與溫度飄移(PVT Variations)的缺點,因此一般而言,超低電壓(Ultra-Low Voltage, ULV)多用於資料傳輸率較低的應用在系統晶片裡,常常有多組 PLLs/DLLs, 提供不同速度的 I/O 界面的工作時脈。因為傳統 PLL/DLL 通常鎖定時間都很 長,因此導致 PLL/DLL 不常關閉,也因此當系統待命時,一直在運作的 PLL/DLL 的功率消耗就變成 SoC 待機(Standby)功率消耗的主要來源。

因此,在本論文中,我們提出一適用於 DVFS 下的低電壓、低耗能、快速 鎖定之全數位鎖相迴路,以降低 SoC 中的關鍵模組: PLL/DLL 的工作電壓並 降低其功率消耗,是本設計主要的目的與貢獻。

本論文所提出之全數位鎖相迴路以 90 奈米標準 CMOS 製程實現,並驗證 所提出的電路架構。

**刷鍵字:**動態電壓與工作頻率調整、全數位鎖相迴路、低耗能、快速鎖定

#### Abstract

In recent years, biomedical electronic applications, such as biological signal monitoring devices, implantable medical devices, and wireless body sensors become more and more popular now. In these battery-powered systems, low energy is a primary concern to increase the system operating time. Therefore, power management is an important issue for designing these devices.

Dynamic voltage and frequency scaling (DVFS) serves an effective means to reduce the dynamic power consumption of the system. Moreover, the duty-cycle control of the power switch can further reduce the standby power consumption of the system.

However, since reducing the supply voltage, circuit will be slowed down, and become sensitive to the PVT Variations. Thus, the ultra-low voltage is usually adopted in the low frequency applications. There are many PLLs / DLLs provide the clock of I/O interface. Traditionally, PLL / DLL usually has long lock-in time. Thus, they can't be turned off for reducing the standby power consumption. When the system is switched to the sleeping mode, the continuous operating PLLs often dominate the standby power consumption of the system.

Therefore, in this thesis, we propose a fast lock-in ADPLL with low power consumption for supporting DVFS scheme.

In addition, the test chip is implemented and verified in 90nm CMOS process with standard cells.

Keywords : Dynamic Voltage and Frequency Scaling (DVFS), All-Digital Phase-Locked Loop (ADPLL), Low Power, Fast Lock-in



I would like to express the deepest gratitude to my advisor, Prof. Ching-Che Chung. He always gives me the right direction and enthusiastic guidance, and helps me to complete my research.

I would like to thank to my partners of Silicon Sensor and System (S3) Lab of National Chung Chen University, too. Without their assistance, I could not overcome many difficulties over two years.

Finally, I would like to thank my family and my girlfriend, who always concern for me. This really makes me to be optimistic every time.

## Content

Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Dynamic Voltage and Frequency Scaling	2
1.3 Design Challenges in Conventional PLLs	3
1.4 Conventional Fast Lock-In Methods Survey	8
1.4.1 Binary Search Algorithm	8
1.4.2 SAR Frequency Search Algorithm	11
1.4.3 TDC Based Fast Lock-in Method	13
1.4.4 Flying Adder Fast Frequency Synthesizer	14
1.4.5 Frequency Estimation Algorithm	15
1.5 Summary	17
1.6 Thesis Organization	18
Chapter 2 All-Digital Phase-Locked Loop for Dynamic Voltage	and
Frequency Scaling	19
2.1 The Proposed Fast-Locked ADPLL Overview	19
2.2 The Proposed Frequency Estimation Algorithm	19
2.3 Summary	26
Charten 2 Cincrit Design and Invelopments tion of ADDI I	
Chapter 3 Circuit Design and Implementation of ADPLL	29
3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I	<b>29</b> Digital
3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter	<b>29</b> Digital 29
3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter	29 Digital 29 30
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	29 Digital 29 30 31
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter.</li> <li>3.1.1 Coarse Tuning Stage</li></ul>	29 Digital 29 30 31 34
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter.</li> <li>3.1.1 Coarse Tuning Stage</li></ul>	29 Digital 29 30 31 34 39
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter.</li> <li>3.1.1 Coarse Tuning Stage</li> <li>3.1.2 Fine Tuning Stage</li> <li>3.1.3 Ratio Parameter Calculation Procedure</li> <li>3.2 Pulse Latch DFF</li> <li>3.3 Phase and Frequency Detector</li> </ul>	<b>29</b> Digital 29 30 31 34 39 43
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	29 Digital 30 31 34 34 39 43 45
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	29 Digital 30 31 34 39 43 45 46
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital29303134394345 <b>46</b>
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital29303134394345 <b>46</b> 4650
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital2930313434454546465053
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital 29 30 31 34 34 45 <b>45</b> <b>46</b> 46 46 50 53 55
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital 29 30 31 34 39 43 45 <b>46</b> 46 46 50 53 55 <b>68</b>
<ul> <li>3.1 Monotonic Digital Controlled Oscillator Embedded Cyclic Time-to-I Converter</li></ul>	<b>29</b> Digital293031343943454646505355 <b>68</b>

ferences
----------



## **List of Figures**

Fig. 1.1: Ultra-low Supply Voltage Receiver Block Diagram[11]	2
Fig. 1.2: Architecture of Charge-Pump Based PLL [12]	3
Fig. 1.3: Forward Body Bias Technology[14]	4
Fig. 1.4: Cross-Section View of the Dynamic Threshold Voltage CMOS	
[3]	5
Fig. 1.5: Cross-Section View of Surface Doping Across the Channel [20]	
	5
Fig. 1.6: Multi-Band Selection Illustration [18]	7
Fig. 1.7: Architecture of Traditional Binary Search Algorithm ADPLL	8
Fig. 1.8: The Traditional Binary Search Algorithm Process	9
Fig. 1.9: The Modify Binary Search Algorithm Process	10
Fig. 1.10: The Gear-Shifting Search Algorithm Process	10
Fig. 1.11: The Successive Approximation Register (SAR) DLL	11
Fig. 1.12: Flowchart of 3-Bits SAR Search Algorithm	12
Fig. 1.13: TDC-Based Fast Lock-In Architecture	13
Fig. 1.14: FA-Based Frequency Synthesizer Architecture	14
Fig. 1.15: FEA-Based Frequency Synthesizer Architecture [29]	15
Fig. 1.16: The measured DNL of DCO [29]	16
Fig. 2.1: The Block diagram of the Proposed ADPLL	19
Fig. 2.2: The Timing diagram of the Proposed ADPLL	54
Fig. 2.3: The Relationship of P, R, and W	23
Fig. 2.4: The Frequency Error Analysis at 445MHz	26
Fig. 2.5: The Frequency Error Analysis at 225MHz	27
Fig. 2.6: The Frequency Error Analysis at 150MHz	27
Fig. 3.1 Proposed DCO Embedded Cyclic TDC	29
Fig. 3.2 Coarse-Tuning Stages Architecture	31
Fig. 3.3 Fine-Tuning Stage of the DCO	32
Fig. 3.4 Fine-Tuning Interpolation Signal Illustration	32
Fig. 3.5 Relationship Between DCO Control Code and DCO Output	
Period	33
Fig. 3.6 The Initial State in TDC is Disabled	34
Fig. 3.7 The Period of the REF_CLK is Smaller than All Coarse Delays	35
Fig. 3.8 The Period of the REF_CLK is Larger than All Coarse Delays	37
Fig. 3.9 The Period of the REF_CLK is Larger than the Maximum	

DCO Clock Period	38
Fig. 3.10 Pulse Latch DFF Architecture	39
Fig. 3.11 PLDFF Data Latch Procedure Illustration	39
Fig. 3.12 Pulse Latch DFF Schematic	40
Fig. 3.13 Layout of Pulse Latch DFF and Standard Cell Library DFF	42
Fig. 3.14: PFD Architecture	43
Fig. 3.15: Digital pulse amplifier architecture	44
Fig. 3.16 Frequency Divider Architecture	45
Fig. 3.17 Phase Error of REF_CLK and OUT_CLK Illustration	45
Fig. 4.1 Layout of the Test Chip	46
Fig. 4.2: Chip Floorplan and I/O Plan.	48
Fig. 4.3 System simulation of proposed ADPLL at TT corner	50
Fig. 4.4 System simulation of proposed ADPLL at FF corner	51
Fig. 4.5 System simulation of proposed ADPLL at SS corner	52
Fig. 4.6 Cycle-to-Cycle Jitter at 640MHz, 1.0V Supply Voltage	53
Fig. 4.7 Period Jitter at 640MHz, 1.0V Supply Voltage	53
Fig. 4.8 Cycle-to-Cycle Jitter at 80MHz, 0.5V Supply Voltage	54
Fig. 4.9 Period Jitter at 80MHz, 0.5V Supply Voltage	54
Fig. 4.10 Microphotograph of the ADPLL	57
Fig. 4.11 Measured Jitter Histogram of Output Clock at 1.0V, 600MHz	
divided by 2	58
Fig. 4.12 Measured Jitter Histogram of Output Clock at 1.0V, 60MHz	60
Fig. 4.13 Measured Jitter Histogram of Output Clock at 0.52V,	
120MHz	62
Fig. 4.14 Measured Jitter Histogram of Output Clock at 0.52V, 30MHz	64

## **List of Tables**

Table 3.1: Comparison Table of Standard Cell Library DFF and PLDFF	
at a 0.5V the Supply Voltage	.41
Table 3.2: The Reset Pulse Constrained of Standard Cell Library DFF	
and PLDFF at a 0.5V Supply Voltage	.44
Table 3.3: The Dead Zone of PFD with Standard Cell Library DFF and	
PFD with PLDFF at a 0.5V Supply Voltage	.44
Table 4.1 Block module name	.47
Table 4.2 I/O PAD Description	.48

Table 4.3 Simulation with PVT Variation at the 1.0V Supply	52
Table 4.4 Simulation with PVT Variation at the 0.5V Supply	52
Table 4.5 Chip summary of Simulation Result	55
Table 4.6 Comparison Table of Simulation Result	56
Table 4.7 Chip summary of Measurement Result	557
Table 4.8 Comparison Table of Measurement Result	568



# Chapter 1

### Introduction

#### **1.1 Motivation**

In recent years, biomedical electronic applications, such as biological signal monitoring devices, wireless body sensors [1] [2], and implantable medical devices [3]-[6], become more and more popular now. In these battery-powered systems, low energy is a primary concern to increase the system operating time. Therefore, power management is an important issue for designing these devices.

Dynamic voltage and frequency scaling (DVFS) serves an effective means to reduce the dynamic power consumption of the system. Moreover, the duty-cycle control of the power switch can further reduce the standby power consumption of the system.

In the system-on-a-chip (SoC), there are several phase-locked loops (PLLs) and delay-locked loops (DLLs) to provide different clock sources for different modules. However, conventional analog charge-pump based phase-locked loops (CP-PLLs) often take a long lock-in time. Thus, they are not possible to be turned off for reducing the standby power consumption. When the system is switched to the sleeping mode, the continuous operating PLLs often dominate the standby power consumption of the system.

As a result, the PLLs which can both operate at a low voltage and have a fast lock-in time are demanded in these applications.

#### **1.2 Dynamic Voltage and Frequency Scaling**

Dynamic voltage and frequency scaling (DVFS) is a mature technology, it has already been widely used in power management on the SoC for saving the dynamic power consumption. Dynamic power consumption occupies approximately 80%-90% of the whole power consumption. Therefore, how to reduce dynamic power consumption is a critical issue [7].

$$P_{dynamic} = \mathbf{a} \cdot C \cdot V dd^2 \cdot f \tag{1.1}$$

As the (1.1),  $P_{dynamic}$  is the dynamic power consumption, *a* is the activity factor, i.e., the switching activities, *C* is the switched capacitance, *Vdd* is the supply voltage, and *f* is the clock frequency. While the supply voltage of circuit is reduced, the dynamic power consumption will be reduced in quadraticlly. Therefore, the most efficient means to reduce the power consumption of system is to decrease the supply voltage.



Fig. 1.1: Ultra-Low Supply Voltage Receiver Block Diagram [11]

Currently, there are many related research papers published [8]-[11]. Fig.1.1 shows the block diagram of a 0.55V IR-UWB Baseband Processor system [11]. We can see that the digital baseband already reduces the supply voltage to 0.55V by the parallel architecture and the sub-threshold circuit technology. However, we also observed that some block circuits such as a low noise amplifier (LNA), a signal mixer,

an analog-to-digital converter (ADC) and a DLL-based multi-phase clock generator which can't reduce the supply voltage with the voltage scaling.

If the supply voltage of PLLs / DLLs can be also adjusted to the lower voltage with voltage scaling, it can eliminate an additional voltage source. In addition, it can also reduce the power network routing complexity.

#### **1.3 Design Challenges in Conventional**

#### **PLLs**



Fig. 1.2: Architecture of Charge-Pump Based PLL [12]

There are some conventional approaches [3], [12]-[19] proposed to implement the low supply voltage PLLs / DLLs. Fig. 1.2 shows the charge-pump based architecture. When the operating voltage is reduced close to the threshold voltage (Vt), drive current (Id) of transistor will rapidly dropped. Thus, most of the low supply voltage charge-pump based PLLs / DLLs often use the forward body bias (FBB) and reverse short channel effect (RSCE) technologies for decreasing the threshold voltage, and then suit to a low supply voltage circuit design.



Fig. 1.3: Forward Body Bias Technology [14]

Fig. 1.3 shows the forward body bias technology. When the forward bias on the body of the PMOS transistor, the threshold voltage (Vt) will decrease. Thus it can increase the driving current at a low supply voltage. However, the forward body bias technology will increase the leakage current, and also increases the static power. As a result, most of the PLLs / DLLs which use the FBB technology, they will implement their design in a process with less leakage problems, such as  $0.25\mu$ m [13]-[15] or  $0.13\mu$ m [3] [16] [17], or just use the silicon on insulator (SOI) process [19] to implement the low supply voltage design.

There are some PLLs / DLLs using 90nm process to achieve low-voltage design [12] [18]. However, they use the bulk driven. By biasing the portion of the bodies of transistors, and not all of the transistors are forward biased. However, if only a portion of the transistors used FBB technology, that it needs to considerate the body noise problems. As mentioned in [3], that the deep N-well must be added to separate the biasing transistors out of the other transistors. As shown in Fig. 1.4 shows, it shows that the triple-well processes must be used to fabricate this transistor instead of using the typical twin-well CMOS process. Thus it will lead to the increasing of the design

cost.



Fig 1.4: Cross-Section View of the Dynamic Threshold Voltage CMOS [3]

Reverse short channel effect (RSCE) [20] technology is usually used in the low supply voltage circuit design. Fig 1.5 illustrates the RSCE. In the advance CMOS process, for preventing drain induced barrier lowering (DIBL) phenomenon and the body punch-through, halo doping is used to implant in the junction between body and source, and the junction between body and drain in the transistor.



Fig 1.5: Cross-Section View of Surface Doping Across the Channel [20]

However, the halo doping will bring an effect that making the doping density increasing at the short channel, and the threshold voltage increase as the doping density. Thus that, if the channel length is increased, the larger distance between the halo doping regions will decrease the doping density of the channel. Then the threshold voltage will decrease as the doping density.

After the threshold voltage is decreased, the transition time of the transistor will be decreased. However, the RSCE technology will increase the area of the transistor since the length of the transistor is increased. In [20], in 0.13µm process, when the transistor length is stretched to over quadruple size, the threshold voltage can't be decreased significantly. As the result, in 0.13µm process, the cost of area may not get enough benefit in the circuit speed.

As the above discussion, in the published papers, there are no low-cost, low-voltage charge-pump based PLL/DLL in 90nm CMOS process. There are several main reasons:

- 1. The leakage power problem become serious in 90nm CMOS process, it is hard to use the FBB technology for the charge-pump based PLL/DLL working at the low supply voltage. Even if the circuit can be realized, it needs to implement with the triple-well process [3], and may increase the cost of chip fabrication. Moreover, the charge-pump circuit design is very limited circuit architecture due to the low supply voltage. Therefore, charge-pump circuit is difficult to ignore the static power consumption. For instance, the static power is 35μW in [12]. When the temperature is increased from room temperature to 100°C, that will increase an order of the static power consumption. Then the control voltage ripple making frequency migration problem will become more serious.
- 2. At low supply voltage, the  $K_{VCO}$  of the voltage-controlled oscillator (VCO) will become very large due to the restricted voltage headroom. Traditionally, the reasonable range of the  $K_{VCO}$  is about 50MHz / V ~ 100MHz / V.

However, in the 0.6V 2.4GHz PLL [15], the  $K_{VCO}$  is 400MHz / V, and in the 0.5V 1.3GHz VCO [13]  $K_{VCO}$  achieves to 2.35GHz/V. The large  $K_{VCO}$  will make the poor tolerance to the control voltage ripples, which means the VCO output frequency is very sensitive to the little noise on the control voltage. Therefore, the current PLLs / DLLs [3] [12] [18] use multi-band technology for reducing the  $K_{VCO}$  by selecting the segmented frequency bands mutually as shown in Fig 1.6. The multi band selection in 0.5V 2.4GHz PLL [12], it is selected according to the user's demand, by the input digital control pins. Besides, there are automatic searching frequency band methods, however, since it needs to search the frequency band in sequentially, the lock-in time will become more longer.



Fig 1.6: Multi-Band Selection Illustration [18]

3. In addition, with the DVFS scheme, the supply voltage is not only at the low-voltage(≤0.5 V), but also needs to switch to the nominal-voltage (1.0V). However, in the VCO of charge-pump based PLLs / DLLs adopting the multi-band technology for increasing the controllable frequency range, the band selection pattern is fixed at the low-voltage. While the supply voltage is

scaled back to the nominal-voltage, the multi-band selector will not build continuous tuning range in the same band selection pattern.

#### **1.4 Conventional Fast Lock-In Methods**

#### Survey

Besides the low-voltage issue, fast lock-in method is also indispensable. All-digital phase-locked loops (ADPLLs) [22]-[29] usually achieve a relatively fast lock-in time than the CP-PLLs, and they can be easily integrated with other digital circuits. Therefore, ADPLL is suitable for biomedical electronic applications.

However, how to overcome the low-voltage design challenges with voltage scaling is a critical issue to conventional ADPLLs. In the following section, we will discuss these issues.

#### 1.4.1 Binary Search Algorithm



Fig 1.7: Architecture of Traditional Binary Search Algorithm ADPLL

Traditional binary search scheme [21] is widely used to search for the target frequency and then reducing the lock-in time of the ADPLLs. As shown in Fig. 1.7, the binary search controller sends the control code (DCO\_code) of the digital controlled oscillator (DCO) to adjust the output frequency (**OUT\_CLK**). According to the lead or lag information between the reference clock (**REF\_CLK**) and the divided feedback clock (**DIV\_CLK**), the controller can control the DCO and achieves

lock.



Fig 1.8: The Traditional Binary Search Algorithm Process

Fig. 1.8 illustrates the frequency searching process. The DCO\_code is set to middle frequency in the beginning. Then, when the frequency of **DIV\_CLK** is lower than the frequency of **REF\_CLK**, the binary search controller will add the tuning step to the DCO\_code for increasing the frequency of **DIV\_CLK**. Oppositely, when the frequency of **DIV\_CLK** is higher than the frequency of **REF\_CLK**, the binary search controller will subtract the tuning step to the DCO\_code for decreasing the frequency of **DIV\_CLK**. In addition, whenever the output of PFD changes from lead to lag or vice versa, the tuning step of DCO\_code is divided by 2. After the tuning step reduced to 1, the frequency searching is finished.

Moreover, in [22], it optimizes the traditional binary search algorithm as a modify binary search algorithm. As shown in Fig. 1.9, whenever the output of PFD changes from lead to lag or vice versa, the DCO\_code will jump to the intermediate between last change DCO code and the current. Thereby, it can furthermore reduce the lock-in time.



Fig 1.9: The Modify Binary Search Algorithm Process

Besides, the Gear-shifting search algorithm is proposed in [24], Fig. 1.10 shows the Gear-shifting search process. Whenever the output of PFD is consecutive leading or laging for several cycles, the DCO\_code in the proportional path is calculated by multiplying by proportional constant. Besides, while the output of PFD changes from lead to lag or vice versa, the DCO\_code will jump to the value calculated by the integral path.



Fig 1.10: The Gear-Shifting Search Algorithm Process

The Gear-shifting algorithm can further enhance the frequency searching ability. However, the proportional constant and integral constant are custom for suiting certain situations. Therefore, while the DVFS manager scale down the supply voltage, the intrinsic delay of delay cell will be changed, and the set of the constants, will be -10different at the situation with the low supply voltage. Thus that, it is not suiting for the voltage scaling with DVFS scheme.

Binary search algorithm is a convenient method and easy to be implemented. However, due to the searching accuracy is determined by the dead-zone of PFD at the low supply voltage, the larger detectable reset pulse width of D-type flip flop (DFF) will become larger. In other words, the detect ability of PFD will become poor. That may cause the binary search algorithm can't lock to the proper target frequency.

#### 1.4.2 SAR Frequency Search Algorithm

In order to improve the lock-in speed, the successive approximation register (SAR) algorithm is used in the DLLs for reducing the lock-in time. For a n-bits SAR-controlled DLL [25], as shown in Fig. 1.11. It only needs n clock cycles to find the optimal delay of delay line and then achieve the lock-in.



Fig 1.11: The Successive Approximation Register (SAR) DLL

An example of 3-bits SAR search algorithm is shown in Fig. 1.12. In the beginning (#1), the most significant bit (MSB) of SAR control code is set to 1. Then, the phase detector (PD) will detect the phase between **REF\_CLK** and - 11 - **OUT\_CLK**. If the result is lead, it represents the phase of **OUT\_CLK** is leading to the phase of **REF\_CLK**, thus the optimal delay can be achieved by setting "0" to the MSB of SAR control code. Then, at #2, the MSB of SAR control code is set to 0, and the second bit of SAR control code is set to 1. After SAR controller repeats the above action until all the bits of SAR control code are detected, at #3, the optimal delay of delay line is found. As the result, 3-bits SAR control code is found after three clock cycles, and achieve the phase lock-in.



Fig 1.12: Flowchart of 3-Bits SAR Search Algorithm

However, the accuracy of SAR search algorithm depend on the binary-weighted delay line. It is need to ensure the multiple with each cell delay time is really doubled. While the DVFS manager scales down the supply voltage, even though the binary-weighted delay line kept as multiple in different delay cell at a nominal-voltage. In the other supply voltage, it is hard to design the binary-weighted delay line. Thus the SAR search algorithm is not suitable for DFVS scheme.

#### **1.4.3 TDC Based Fast Lock-in Method**

For fast lock-in methods [26], time-to-digital converter (TDC) is used to quickly calculate the nearest control code for the DCO to produce the desired frequency. TDC can convert the period information of the reference clock into the multiples of the delay time of the delay cell. Hence, the controller can use the period information to adjust the DCO to produce target frequency very quickly.



Fig 1.13: TDC-Based Fast Lock-In Architecture

As shown in Fig. 1.13, the delay cell ring begins to oscillate while the first **REF\_CLK** triggered. Then, the **OUT\_CLK** of delay cell ring triggers the digital processor to count up until the second rising edge of **REF\_CLK**. Finally, the period information of **REF\_CLK** is obtained. According to the period information, the digital processor sends the optimal control code to the path selector for adjusting the DCO to produce target frequency.

In [26], the TDC-based fast lock-in method only takes 7 clock cycles to achieve frequency and phase lock-in. However, in order to avoid the bit number of counter in digital processor being too large, the delay time of delay cell ring is set to be longer. As a result, the resolution of DCO becomes very poor (170ps). This situation will become further serious in the low supply voltage. Thus the TDC-based fast lock-in method is not suitable with DVFS scheme.

#### **1.4.4 Flying Adder Fast Frequency Synthesizer**



Fig 1.14: FA-Based Frequency Synthesizer Architecture

Flying adder (FA) is proposed in [27] [28]. The FA-based frequency synthesizer uses a set of multi-phase signals to compose a desired frequency, and achieve fast frequency lock-in. As shown in Fig. 1.14, the pre-locked PLL locked at first, then the locked-clock is passed to the delay chain, the delay chain is built by a series of delay cells. It can produce the multi-phase locked-clock to compose the desired frequency.

For an example, the delay disparity between each multi-phase is the resolution of FA, it is equal to the propagation delay of the delay cell on delay chain. In [28], the resolution is 0.2625ns. If the FA needs to produce a 380MHz clock, the period of this clock is 2.625ns and is equal to 10 times of the resolution, thus the FA chooses the 1<sup>st</sup> multi-phase and the 10<sup>th</sup> multi-phase. By trigged the rising edges of two chosen multi-phase, FA can generate the desired frequency.

However, frequency synthesizer is an open loop clock generator. Hence, the pre-locked PLL circuit can't be shut down. Due to the output frequency is synthesized by composing the multi-phase from the pre-locking PLL. If the circuit restarts, the frequency synthesizer needs to wait a long time for the PLL re-locked. Thus that, the FA-based frequency synthesizer is not suitable with DVFS scheme.

#### **1.4.5 Frequency Estimation Algorithm**

Frequency estimation algorithm (FEA) is proposed in [29]. It utilizes the linear characteristics of three DCOs' output period and fractional factor to calculate the desired DCO control code. Then, it can achieve fast lock-in time.

The architecture of FEA-based ADPLL is shown in Fig. 1.15. Part (a) is the fast lock-in block. That includes two inner DCOs, two frequency counters, and a DCO parameter calculator. When the system is reset, the inner  $DCO_H$  is set to generate the fastest frequency, and the inner  $DCO_L$  is set to generate the slowest frequency. At the same time, there are two period ratio parameters obtained by using the two frequency counter to quantize the fastest frequency and slowest frequency by the reference clock, respectively. Finally, by the DCO parameter calculator, the linear period curve of output frequency is built, and the DCO code for the desired frequency can be calculated, and sent to the outer DCO in part (b).



Fig 1.15: FEA-Based Frequency Synthesizer Architecture [29]

This method not only achieves fast lock-in time but also has the PVT variations tolerance. Since the period ratio parameters obtained from the linear frequency curve varies with the PVT variations.

However, the frequency estimation algorithm needs a fine resolution and a high linearity DCO with monotonic response. Moreover, the ADPLL [29] requires a large frequency multiplication ratio to reduce the quantization effect of the frequency counter. Otherwise, the target DCO control code calculated by the proposed equation will still have a large frequency error after two clock cycles.

In addition, the ADPLL [29] uses a frequency counter to obtain the require cycle count information. Thus, the cycle time ratios between the reference clock and the DCO are integer numbers which cause considerable calculation errors in the equation. Moreover, the proposed tri-state inverter-based DCO does not have a fine-tuning stage, besides, the linear characteristics of the output frequency curve is very poor. The differential non-linearity (DNL) of this DCO is shown in Fig. 1.16. As the result, the DNL is larger than one LSB, it represents the output clock of DCO doesn't have a good linearity characteristics. Thus, it cannot achieve the wide-range operation.



Fig 1.16: The measured DNL of DCO [29]

Besides, it needs three DCOs to implement the FEA [29], the period curves in these three DCOs are not easy to be kept the same. It also brings the calculation error for the FEA due to the on-chip variations. As the result, the frequency range of the DCO is very limited. Thus, it is not suitable for biomedical electronic applications with DVFS scheme.

#### **1.5 Summary**

In the above low-voltage and fast lock PLL architectures, they are not suitable for biomedical electronic applications with DVFS scheme. Therefore, in this thesis, we propose an FEA-based ADPLL for battery-powered devices with DVFS scheme. The proposed ADPLL can work and achieve fast lock-in at nominal supply voltage (1.0V) and low supply voltage (0.5V).

The proposed frequency estimation algorithm with an embedded cyclic TDC can quickly calculate the target DCO control code with a high precision. Thus, the proposed ADPLL can achieve fast lock-in time in four clock cycles.

We adopt the FEA-based architecture and implement the proposed ADPLL in cell-based design flow. Although the transition time and the propagation delay of the logic gates are increased with a low supply voltage. Most logic gates of the standard cell library can still work correctly at a low supply voltage. However, the sequential elements (i.e. D-type flip-flops, DFFs) often have unacceptable setup time and hold time margins and have a large clock-to-Q delay at a low supply voltage. In the ADPLL, the DFFs of the frequency divider will operate at the maximum frequency of the DCO. As a result, a smaller clock-to-Q delay with narrower setup time and hold time margins are needed for design the DFFs.

In addition, the DFFs of the TDC also require a narrower time margin for reducing the sample error. The DFFs of the phase and frequency detector (PFD) also needs a smaller time margins for reducing the dead zone. As a result, the design of the DFFs at a low supply voltage is very important. The FBB technique with a true single phase clock (TSPC) DFF [30] or the pulse-latch DFF [31] can help to build up the DFF for the low-voltage ADPLL.

#### **1.6 Thesis Organization**

In this paper, the rest of the paper is organized as follows. Chapter 2 presents the proposed ADPLL architecture and the proposed frequency estimation algorithm. Chapter 3 discusses the circuit implementation, include the monotonic DCO, an cyclic TDC embedded DCO, the pulse latch DFF (PLDFF), the PFD and the frequency divider with PLDFF. Chapter 4, measurement and simulation results of the proposed ADPLL are discussed. Finally, in Chapter 5, we make a conclusion of this thesis and describe the further works about several design issues which can be extended in the further.



#### **Chapter 2**

## All-Digital Phase-Locked Loop for Dynamic Voltage and Frequency Scaling



Fig. 2.1: The Block diagram of the Proposed ADPLL

The block diagram of the proposed ADPLL is shown in Fig. 2.1. The ADPLL is composed of a phase and frequency detector (PFD), an ADPLL controller with a

digital loop filter (DLF), a frequency finder (FF), a monotonic low-power cyclic TDC-embedded DCO, and a frequency divider.

After system is reset, the PFD and the frequency divider are stopped waiting for the frequency finder to calculate the initial DCO control code (init\_code) for the ADPLL controller. We adopt a similar FEA-based fast lock-in method to produce the desired DCO control code. However, in [29], there are many drawbacks needed to be overcome. Such as the resolution and linear characteristic of DCO are very poor, and a quantization error by using the frequency counter to obtain the period ratio parameters. Besides, the transition time and the propagation delay of the sequential elements will be increased with a lower supply voltage.

Therefore, we propose a monotonic DCO embedded cyclic-TDC to solve the problem of the resolution and linear characteristic of DCO, and reducing the quantization error of the period ratio parameters. Besides, the DCO adopt the interpolator-based fine-tuning stage, it keeps the monotonic response when the DCO control code switches cross over different coarse-tuning control codes either at 0.5V and 1.0V. In addition, the fine-tuning stage can always provide a total delay tuning range equal to one coarse-tuning resolution under process, voltage, and temperature (PVT) variations.

Moreover, we adopt the pulse latch DFFs (PLDFFs) to replace the DFFs of the standard cell library. It can improve the setup time, hold time, and clock-to-Q delay of the DFFs at a low supply voltage.

Then, the PFD and the frequency divider are enabled. The proposed ADPLL can achieve fast lock-in in four clock cycles. Subsequently, the ADPLL controller updates the DCO control code (DCO\_code) according to the PFD's output to keep tracking the phase and frequency of the reference clock.

Since the frequency divider operates at the maximum frequency of the DCO, a \$-20\$-

smaller clock-to-Q delay of the PLDFF is necessary. In addition, the PLDFF is used in the embedded cyclic TDC and the PFD. The PLDFFs of the TDC require a narrower time margins for reducing the quantization error. Besides, the PLDFFs of the PFD also need to have a smaller time margins for reducing the dead zone. As a result, the design of the PLDFFs at a low supply voltage is very important.

Finally, the digital loop filter [22] is applied to produce the baseline DCO control code (avg\_code) for reducing the reference clock jitter effects.

#### **2.2 The Proposed Frequency Estimation**



#### Algorithm

Fig. 2.2: The Timing diagram of the Proposed ADPLL

The timing diagram of the proposed frequency estimation algorithm is shown in Fig. 2.2. The period of the DCO output clock (**OUT\_CLK**) is a function of the DCO control code (**DCO\_code**), and is named as P(code). When the DCO control code is set to zero, the period of the DCO is at its maximum value ( $P_{max}$ ), as shown in Fig. 2.3(a). Oppositely, when the **DCO\_code** is set to 2047, the period of the DCO is at its minimum value ( $P_{min}$ ). The period ratio between the reference clock (**REF\_CLK**) and the DCO clock (**OUT\_CLK**) is also a function of the **DCO\_code**, and is named as R(code). The  $R_{max}$  and  $R_{min}$  mean that the period ratios when the DCO operates at  $P_{min}$  and  $P_{max}$ , respectively. The value of  $R_{max}$  and  $R_{min}$  are expressed in Eq. (2.1) and Eq. (2.2):











(c)

Fig. 2.3: The Relationship of P, R, and W

In [29], the ratio R(code) is obtained by using a frequency counter. Therefore, the value of the R(code) will be an integer number, and it has quantization effects, as shown in Fig. 2.3(a). However, if the value of R(code) is still obtained by a frequency counter, that will bring the quantization effect. If we define a new function W(code) which is the reciprocal of the R(code), and then, the value of the W(code) will be a

fixed-point number. However, if the value of R(code) is still obtained by a frequency counter, the quantization error of W(code) will be very large, as shown in Fig.2.3(b).

In this thesis, we uses a cyclic TDC to obtain the fixed-point value of the R(code). Thus, the quantization effects of W(code) can be significantly reduced, as shown in Fig. 2.3(c). Eq. 2.3 and Eq. 2.4 show the values of  $W_{max}$  and  $W_{min}$ :

$$W_{max} = \frac{1}{R_{min}}$$
(2.3)

$$W_{min} = \frac{1}{R_{max}}$$
(2.4)

Then, the W(code) curve between  $W_{max}$  and  $W_{min}$  will be a straight line. Therefore, the equation of W(code) can be expressed as Eq. 2.5.

$$W(code) = \frac{W_{max} - W_{min}}{2^{11} - 1} \times code + W_{max}$$
(2.5)

Since the frequency multiplication factor (**M**) is an input value, the target period ratio (**R**<sub>T</sub>) is equal to **M**. Then, the value of the target W(init\_code) is equal to  $W_T=1/M$ . If the constant value (2<sup>11</sup>-1) in the Eq. 2.6 is reduced as 2<sup>11</sup>, the target DCO control code (init\_code) can be found using Eq. 2.6.

$$init\_code = 2^{11} \times \frac{W_{max} - W_T}{W_{max} - W_{min}}$$
(2.6)

In the proposed ADPLL, one divider is used to calculate the value of  $W_{max}$ ,  $W_{min}$ , and the initial code of the DCO (init\_code) in three clock cycles. In Fig. 2.2, after system is reset, in the first cycle, the state of the ADPLL controller is **R\_MIN**, and in this cycle, the DCO control code is set to zero to use the proposed cyclic TDC for calculating the value of **R**<sub>min</sub>. In the second cycle, the state is changed to **R\_MAX**, the DCO control code is set to 2047 for calculating the value of **R**<sub>max</sub>. Meanwhile, the constant value 2<sup>19</sup> and **Rmin** are sent to the divider for calculating the **W**<sub>max</sub>.
In the third cycle, the state is changed to **PIPE\_1**, the constant value  $2^{19}$  and **R**<sub>max</sub> are sent to the divider for calculating the **W**<sub>min</sub>. In the fourth cycle, the state is changed to **PIPE\_2**, and the initial code of the DCO (init\_code) is calculated using Eq. 2.6.

In the ADPLL [29], three DCOs are used for calculating the required parameters. However, these DCOs may have on-chip variations (OCVs) especially in advanced CMOS process. In addition, the integer values of  $\mathbf{R}_{min}$  and  $\mathbf{R}_{max}$  are used in the ADPLL [29] for calculating the target DCO control code which results in large quantization errors. Therefore, the ADPLL [29] still have a large frequency error after two clock cycles. The proposed ADPLL uses a monotonic low-power cyclic TDC-embedded DCO to avoid on-chip variations, and we use fixed-point values of  $\mathbf{R}_{max}$  and  $\mathbf{R}_{min}$  with the proposed frequency estimation algorithm for calculating an accurate target DCO control code (init\_code). As a result, the proposed ADPLL can achieve a relatively small frequency error after four clock cycles.

1

#### 2.3 Summary

As compares to [29], the proposed frequency estimation algorithm is further suitable for the DVFS scheme. Since we enhance the precision of the period ratio parameters and the linear characteristic by the proposed monotonic DCO embedded cyclic TDC. Besides, although the [29] can achieve lock in two clock cycles, it still has a large frequency error due to the on-chip variations of three DCOs and the quantization effects of the frequency counter. The proposed ADPLL has a relatively fast lock-in time.



Fig. 2.4: The Frequency Error Analysis at 445MHz

As shown in Fig. 2.4, we compare the frequency error of the FEA between the [29] and the proposed ADPLL with 5MHz reference clock, and the multiplication factor is equal to 89. The frequency error is larger than 20% by FEA of the [29] in 2 clock cycles, and it needed to keep tuning the DCO control code by the binary search algorithm and takes more than 40 clock cycles for fine-tuning the output frequency to the target frequency.

In the proposed FEA, the frequency error is about 3% after 4 clock cycles, due to the clock skew between the **ENABLE\_DCO** and **LOCK** as shown in Fig 2.2, there is a phase error between the 4<sup>th</sup> reference clock and the first output clock. Thus that, we need to align the phase of reference clock and output clock after FAE. After 40 clock



cycles, the frequency error will be decreased to less than 1.5% by the digital loop filter.

Fig. 2.5: The Frequency Error Analysis at 225MHz

As shown in Fig. 2.5, we compare the frequency error of the FEA between the [29] and the proposed ADPLL with 5MHz reference clock, and the multiplication factor is equal to 45. The frequency error is larger than 30% by FEA of the [29] in 2 clock cycles, and it needs to keep fine-tuning output frequency with the binary search algorithm for more than 50 clock cycles.

That means the precision of FEA is sensitive to the  $\mathbf{R}_{min}$  as shown in Fig. 2.3, due to there is a quantization error at the  $\mathbf{R}_{min}$ , thus if the multiplication factor is close to the  $\mathbf{R}_{min}$ , the frequency error of FEA become further larger. Hence, in the [29], it has a constraint of the multiplication factor to be larger than 45.



Fig. 2.6: The Frequency Error Analysis with 150MHz

As shown in Fig. 2.6, we compare the frequency error of the FEA between the -27 -

[29] and the proposed ADPLL in 5MHz reference clock, and the multiplication factor is equal to 30. The frequency error is larger than 35% by FEA of the [29] in 2 clock cycles, and it needs to keep fine-tuning output frequency with the binary search algorithm for more longer than 50 clock cycles for approaching the output frequency to the target frequency.

$$F = \frac{T \times R_{min}}{N} \times \frac{R_{max} - N}{R_{max} - R_{min}}$$
(2.7)

Eq. 2.7 shows the frequency estimation algorithm of [29], **F** is the target of DCO control, **N** is the multiplication factor, **T** is the total stages of the DCO control code. Thus, we use only one DCO and has a smaller area cost as compared to [29]. Besides, for the calculator circuit of the target DCO control code, as compare with the Eq. 2.7 [29], we only use a bit shifter instead of the multiplier, and an arithmetic divider. However, in Eq. 2.7 [29], there are two multipliers and two arithmetic dividers are used for calculating the target DCO control code. In addition, for the accuracy of the target DCO control code, they need to perform the expression with larger operand size to realize the calculator circuit, and it increases the chip area.

In addition, in [29], for increasing the frequency range and linear characteristic, it adopts the tri-state inverter array to realize the DCO. This further increases the chip area cost, and is not suitable for voltage scaling. We adopt the interpolator-based fine tuning circuit [32] to overcome the drawback. The interpolator fine tuning circuit can achieve monotonic response either in the nominal-voltage and low-voltage. Besides, it can enhance the resolution of DCO effectively, and then enhances the resolution of the period ratio curve.

As a result, the proposed ADPLL is suitable for biomedical electronic applications with DVFS scheme.

## **Chapter 3**

# **Circuit Design and Implementation** of **ADPLL**

In this chapter, we explain the circuit design in the proposed ADPLL. Including the monotonic DCO embedded cyclic TDC, the pulse latch DFF (PLDFF), and the phase and frequency detector (PFD) and frequency divide with PLDFFs.

### **3.1 Monotonic Digital Controlled Oscillator**

## **Embedded Cyclic Time-to-Digital**

#### Converter





Fig. 3.1 shows the monotonic DCO embedded cyclic TDC architecture. The DCO is composed of 64 coarse tuning stages and 32 fine tuning stages and the cyclic TDC [33] for ratio parameter calculation. A coarse tuning stage is composed of three NAND gates. The fine tuning circuit is composed of two parallel connected tri-state buffer arrays operating as an interpolator [32]. The interpolator circuit can keep the monotonic response between two coarse tuning stages switching either at nominal-voltage and at low-voltage. The cyclic TDC is composed of 64 pulse latch DFFs (PLDFFs) [31] at every output node of two NAND gates. The Encoder encodes the binary 11 bits dco\_code[10:0] into the coarse-tuning and fine-tuning control thermometer code. The dco\_code[10:5] encodes into coarse[62:0] and the dco\_code[4:0] encodes into fine[30:0].

The Decoder decodes the 65-bits thermometer code generated by the cyclic TDC into the 7-bit fractional number and the 9-bit integer number from the cyclic counter of the ratio parameter to the frequency finder.

#### **3.1.1 Coarse Tuning Stage**

The coarse tuning stages of DCO is shown in Fig. 3.2, the coarse-tuning stage consists of 64 coarse tuning delay cells (CDCs). The coarse-tuning resolution of the proposed DCO is two NAND gates delay time.

The NAND bridge based architecture has smaller coarse-tuning resolution than the conventional MUX type DCO [32]. The one coarse tuning stage delay of the conventional MUX type DCO is the sum of a NAND gate delay and a MUX delay. In the proposed coarse tuning architecture, the one coarse tuning stage delay is two NAND gates delay time. As we discuss in section 2.2, we use the two coarse tuning stages as a TDC resolution to quantize the reference clock period. The resolution of a coarse tuning stage should be kept as small as possible. Thus, we can achieve the high precision of the period ratio parameters acquisition for the proposed frequency estimation algorithm.

As shown in Fig. 3.2, if the coarse DCO control code is 61, it should be encoded into the coarse control code as  $63^{\circ}hfff_{fff}_{fff}_{fff}_{fff}$  (coarse[0]~coarse[2] = 0, coarse[3]~coarse[62] = 1). In addition, one coarse tuning delay is equal to 32 times fine tuning delay.



#### **3.1.2 Fine Tuning Stage**

The fine-tuning stage of the DCO is shown in Fig. 3.3. The fine-tuning stage [32] is composed of two parallel connected tri-state buffer arrays operating as an interpolator. When there are more left-hand side tri-state buffers turned on, the output clock is more close to the **CA\_OUT**, and when there are more right-hand side tri-state buffers turned on, the output clock is more close to the **CB\_OUT**. In addition, the timing difference between **CA\_OUT** and **CB\_OUT** is one coarse-tuning resolution. Therefore, the proposed fine-tuning stage can always provide a total delay tuning range equal to one coarse-tuning resolution under process, voltage, and temperature (PVT) variations.



Fig. 3.3: Fine-Tuning Stage of the DCO

As shown in Fig. 3.4, if the fine DCO control code is 5'd29, it should be encoded into the coarse control code as 31'h1fff\_ffff (Fine[0] ~ Fine[28] = 1, Fine [29]~ Fine [30] = 0). The interpolation signal is shown in Fig. 3.4.



Fig. 3.4 Fine-Tuning Interpolation Signal Illustration



(a) Full DCO Control Code to DCO Output Period Simulation



(b) Fine Tuning Code Cross Over 3 Coarse Stages



Fig. 3.5 shows the simulation results of the proposed DCO. The resolution is about 17.39ps and 3.19ps at 0.5V and 1.0V, respectively. Besides, in the Fig 3.5 (b), the proposed DCO keeps the monotonic response when the DCO control code switches cross over different coarse-tuning control codes either at 0.5V and 1.0V.

#### **3.1.3 Ratio Parameter Calculation Procedure**

In chapter 2, we introduce the proposed frequency estimation algorithm. It adopts embedded cyclic TDC to obtain the parameters of period ratio between reference clock and DCO output clock. Then, it builds the linear function by the ratio parameters for calculating the target DCO control code. In the following section, we will explain the ratio parameter obtained procedure.



Fig. 3.6: The Initial State in TDC is Disabled

As shown in Figs. 2.2 and 3.6, when the **Enable\_TDC** is set to "0" to gate the **TDC\_CLK**, all the coarse-tuning cells are turned on for calculating the period ratio parameter ( $\mathbf{R}_{min}$ ) between the **REF\_CLK** period and the maximum DCO clock period. Besides, the **Enable\_DCO** is set to "0" in the beginning, thus that, the DCO will not oscillate, and all the signals on each two of CDC are "0". Besides, the integer number of period ratio parameter (TDC\_Int [8:0]) and the fractional number (TDC\_Fac [6:0]) are both set to 0.

Then, the Enable\_TDC will be set to "1" as shown in Fig. 2.2, and it starts the

ratio parameter calculation procedure, and **Enable\_DCO** is set to "1" for half of the **REF\_CLK** period. The next positive edge of **REF\_CLK** will trigger the PLDFFs deployed on the DCO delay line to obtain the period of **REF\_CLK** in terms of two coarse tuning delays.

In addition, as shown in Eq. 2.2, the period ratio parameter ( $\mathbf{R}_{min}$ ) is equal to the period ratio between the period of **REF\_CLK** and the maximum period of **OUT\_CLK**. However, in order to pipe the calculation procedure in 4 clock cycles, we only use a half cycle of the **REF\_CLK** to calculate the period ratio parameter. Thus, after we obtain the period ratio parameter, its value needs to be multiplied by 2.



Fig. 3.7 The half period of the REF\_CLK is smaller than all coarse delays

In Fig. 3.7, the pulse width of the **TDC\_CLK** is smaller than all coarse-tuning delay line delay. After the **Enable\_TDC** and **Enable\_DCO** are both set to "1", the DCO is beginning to oscillate, and the "1" on the **OUT\_CLK** is propagated to the

NAND gates chain. After a half cycle of **REF\_CLK**, the **TDC\_CLK** will sample the signal on the CDCs by the PLDFFs as LAT\_code [63:0]. The Decoder must to find the farthest "0", and decodes into corresponding fractional number of period ratio parameter. In Fig. 3.7, LAT\_code [63:0] is equal to 64'h0000\_0001\_ffff\_ffff, and it should be decoded as 33<sub>10</sub> to TDC\_Frac [5:0].

The total coarse-tuning delay line delays are equal to half cycle of the maximum DCO output clock period. Therefore, we need to detect the **OUT\_CLK** by LAT\_code [64]. In Fig. 3.7, the value of LAT\_code [64] is "1", that represents the obtained signal is still in the first half of the DCO clock cycle, thus the MSB of the period ratio fractional number TDC\_Frac [6] is equal to 0.

In this sample, the period ratio fractional number (TDC\_Frac [6:0]) is equal to 7'h21, and the integer number (TDC\_Int [8:0]) is equal to 9'h000. Finally, since the period ratio is sampled by a half cycle of **REF\_CLK**, and its value needs to be multiplied by 2, thus the TDC\_code [16:0] is equal to  $\{9'h000,7'h21,1'b0\}$ , i.e.,  $66/128 = 0.516_{10}$ .



Fig. 3.8: The half period of the REF\_CLK is larger than all coarse delays

In Fig. 3.8, the half period of the **REF\_CLK** is larger than all coarse-tuning delay line delays. The "0" should be in front of the NAND gates chain. The Decoder must find the farthest "1", and decodes into corresponding fractional number of period ratio parameter. In Fig. 3.8, the TDC latched code (LAT\_code [63:0]) is equal to 64'hffff\_ffff\_ffff\_fffc, and it should be decoded as  $2_{10}$  to TDC\_Frac [5:0]. Besides, since the value of LAT\_code [64] is "0", it represents the obtained signal is in the second half DCO clock cycle, thus the MSB of the period ratio fractional number TDC\_Frac [6] is equal to 1.

In this sample, the period ratio fractional number (TDC\_Frac [6:0]) is equal to 7'h42, and the integer number (TDC\_Int [8:0]) is equal to 9'h000. Finally, since the period ratio is sampled by a half cycle of **REF\_CLK**, and its value needs to be multiplied by 2, thus the TDC\_code[16:0] is equal to {9'h000,7'h42,1'b0}, i.e.,

 $132/128 = 1.031_{10}$ .



Fig. 3.9: The half period of the REF\_CLK is larger than the maximum DCO clock period

In Fig. 3.9, the period of **REF\_CLK** is larger than the two times of all coarse-tuning delay line delay. In this case, the cyclic counter is adopt to count the positive edge of **CLK\_OUT** for the integer number of period ratio parameter. If the cyclic counter is added, represents that the period of obtained signal is larger than all coarse stages delays. The "1" should be in the front NAND gates chain again. The Decoder must find the farthest "0", and decodes into corresponding fractional number of period ratio parameter. In Fig. 3.9, the TDC latched code (LAT\_code [63:0]) is equal to 64'h0000\_0000\_0000\_0003. It should be decoded as  $2_{10}$  to TDC\_Frac [5:0], and the value of LAT\_code [64] is "1", thus the MSB of the period ratio fractional number TDC\_Frac [6] is equal to 1.

Besides, the cyclic counter is triggered for one times, thus the period ratio integer - 38 -

number (TDC\_Int [8:0]) is equal to 1. In this sample, the period ratio fractional number (TDC\_Frac [6:0]) is equal to 7'h02, and the integer number is equal to 9'h001. Finally, the TDC\_code[16:0] is equal to  $\{9'h001,7'h02,1'b0\}$ , i.e.,  $260/128 = 2.031_{10}$ .

### **3.2 Pulse Latch DFF**



Fig. 3.10 shows the pulse latch DFF (PLDFF) architecture. In Fig. 3.11, when the PLDFF is triggered, the input data D will be evaluated in the implicit pulse at first. The  $T_{pulse}$  is the difference of CLK and the delayed CLK<sub>B</sub> generated by the pulse generator. Then, by the interlaced on-off tri-state latch, the input data D will be latched. By this way, the PLDFF save a pre-charge time as compares with the conventional DFF. Thus the setup time of PLDFF will be very small.







Fig. 3.12: Pulse Latch DFF Schematic

Pulse latch DFF has the advantages of fast response, no pre-charge phase, no current fighting during data latching, high data glitch immunity and stacked structure to minimize leakage, and the setup time approaches to zero. These features can help in TDC obtaining a higher accuracy period ratio parameter. Besides, it can help the other circuits such as PFD and divider to achieve a good performance at the low supply voltage.

Table 3.1: Comparison Table of Standard Cell Library DFF and PLDFF at a 0.5V

Table (ps)	Rise time	Rise delay	Fall time	Fall delay	Setup time	Hold Time
DFFRX1	188	705	138	776	317	50
DFFRXL	186	702	115	727	331	50
PL DFF	64	149	135	300	< 10	150

the Supply Voltage

As shown in Table 3.1, the PLDFF has a better performance than the DFFs in standard cell library at a low supply voltage. Especially, the response time like rise time and rise delay are both better than the DFFs in standard cell library. Besides, since the smaller setup time margin, it can help in TDC obtaining a higher accuracy period ratio parameter.





Fig. 3.13 Layout of Pulse Latch DFF and Standard Cell Library DFF

Fig. 3.13 shows the layout of the PLDFF and DFFs of standard cell library. As compare to the cell size of the DFFs of the standard cell library, PLDFF has two times area than the DFFs of the standard cell library. Besides, the number of PLDFF in proposed ADPLL is 80. Thus PLDFFs will increase the area cost of the proposed ADPLL.

#### **3.3 Phase and Frequency Detector**

Fig. 3.14 shows the PFD architecture. The PFD [21] is designed with standard cells and PLDFFs. It can detect lead or lag information of reference clock and divided clock. If the PFD is triggered earlier by reference clock, and that means the frequency of reference clock is faster than the divided clock. In this case, the UP signal is generated with a low pulse to send UP signal to the ADPLL controller. Oppositely, if the divided clock triggers the PFD earlier, than the DOWN signal will be generated, and sends to the ADPLL controller to slow down the DCO output clock.



Fig. 3.14: PFD Architecture

After detecting lead or lag information and generating UP or DOWN signal, the UP/DOWN signals will be sent to the ADPLL controller. Due to the delay of circuit transition time, the PFD have the dead zone once the phase error between reference clock and divided clock is very small. If the phase error is too small, then the pulse signals OUTU and OUTD will not be detected by the ADPLL controller correctly. Therefore, the pulse amplifier [21] is adopted to extend the pulse as shown in Fig. 3.15.



Fig. 3.15: Digital pulse amplifier architecture.

Besides, the resettable pulse width of the DFF will influence the dead zone of the PFD. Thus we adopt the PLDFF to reduce the dead zone of the PFD. Table 3.2 shows the minimum resettable pulse width of the PLDFF and the DFFs of the standard cell library at a low supply voltage. As the result, the minimum resettable pulse width of the PLDFF is smaller than the DFFs of the standard cell library. In the Table 3.3, it shows the dead zone of the PFD with PLDFFs and the DFFs of the standard cell library. Thus, the PLDFF can reduce the dead zone of PFD at a low supply voltage effectively, it reduces the dead zone of the PFD from 107ps to 26ps.

Table 3.2: The Reset Pulse Constrained of Standard Cell Library DFF and

PLDFF at a 0.5V Supply Voltage

Table (ps)	DFFRX1	DFFRXL	PLDFF
Resettable Pulse Width	>430	>450	>110

Table 3.3: The Dead Zone of PFD with Standard Cell Library DFF and PFD with

PLDFF at a 0.5V Supply Voltage

Table (ps)	Standard Cell Library DFF	PLDFF
Dead Zone	107	26

### **3.4 Frequency Divider**



Fig. 3.16 Frequency Divider Architecture

Fig 3.16 shows the frequency divider architecture. The frequency divider is composed of multi-stages self-triggered PLDFF. Each stage is a divide-by-2 divider. By cascading the self-triggered PLDFF stages, it can divide the **OUT\_CLK** to **DIV\_CLK** by  $2^n$ , where n is the stage number.

Due to the worse rise time and rise delay of the DFFs of the standard cell library, it produces a large skew between **DIV\_CLK** and **DCO\_CLK**. As shown in Fig. 3.17, since the PFD detects the phase difference between **REF\_CLK** and **DIV\_CLK**. If there has a skew between **OUT\_CLK** and **DIV\_CLK**, that will cause a phase error.



Fig. 3.17 Phase Error of REF\_CLK and OUT\_CLK Illustration

## **Chapter 4**

## **Experimental Results**

## 4.1 Simulation Result



Fig. 4.1 Layout of the Test Chip

Block Number	Module Name
(1)	Frequency Finder
(2)	ADPLL Controller
(3)	Digital Loop Filter (DLF)
(4)	Phase and Frequency Detector (PFD)
(5)	Embedded Cyclic TDC
(6)	Test Divider
(7)	Digital Controlled Oscillator (DCO)
(8)	Frequency Divider

Table 4.1 Block module name

The proposed fast lock-in ADPLL is implemented in TSMC 90nm CMOS process. The chip layout is shown in Fig. 4.1. The active area is  $250 \times 250 \ \mu\text{m}^2$  and the chip area is  $864 \times 864 \ \mu\text{m}^2$ . The block module name of each block is shown in Table 4.1. The simulated power consumption is 1.51mW at 640MHz operation frequency at a 1.0V supply voltage. The power consumption is 56.9 $\mu$ W at 160MHz operation frequency at a 0.5V supply voltage.



Fig. 4.2: Chip Floorplan and I/O Plan.

Input	Bits	Function		
CLK_REF	1	Input clock		
RST_B	1	Set chip to initial at 0		
		Set the multiplication	factor of ADPLL	
		Value	Factor code	
		3'd0	2	
DIV M	3	3'd1	4	
		3'd2	8	
		3'd3	16	
		3'd4	32	
		3'd5	64	

		3'd6	128		
		3'd7	256		
		Set the multiplication factor of output clock			
		Value	Factor code		
тест м	2	2'd0	1		
IESI_W	2	2'd1	2		
		2'd2	4		
		2'd3	8		
		Set the number of bypa	assed divisor bit		
	3	Value	Bypassed bits		
		3'b000	0		
DIV_BYPASS		3'b001	1		
		3'b011	2		
		3'b111	3		
Output	Bits	Fun	ction		
CLOCK_OUT	1	Divided output clock			
CLOCK_DIV	1/	Feedback clock			
UP	1	CLOCK_DIV lead to CLOCK_OUT			
DN	1	CLOCK_DIV lag to CLOCK_OUT			
FINIISH_TDC	1	TDC is gated at 0			
ENABLE_DCO	1	DCO is enabled at 1			
ENABLE_DIV	1	Output divider is enabled at 1			
LOCK	1	Phase locked signal			

### **4.2 Full Chip Simulation**



(b) Post-Sim at 0.5V, 160MHz, TT corner

Fig. 4.3 System simulation of proposed ADPLL at TT corner

The proposed ADPLL is implemented with standard cells and PLDFF. The post-layout simulation result of the proposed ADPLL in TT corner is shown in Fig. 4.3. After system is reset, the proposed frequency finder computes the target DCO control code in four clock cycles. After that, the frequency divider and the PFD are turned on, and the proposed ADPLL keeps tracking the phase and frequency of the reference clock. As the result, the proposed ADPLL can achieve fast lock in four cycles. Besides, we simulate the proposed ADPLL with PVT variations, as shown in Fig. 4.4 and Fig. 4.5.



#### (b) Post-Sim at 0.5V, 320MHz, FF corner





#### (a) Post-Sim at 1.0V, 320MHz, SS corner



(b) Post-Sim at 0.5V, 80MHz, SS corner

Fig. 4.5 System simulation of proposed ADPLL at SS corner

Parameter	Post-sim(FF)	Post-sim(SS)	Post-sim(TT)			
Power supply (V)	1.1	0.9	1.0			
Total Cumont (m A)	2.45	0.69	1.51			
Total Current (mA)	(@640MHz)	(@320MHz)	(@640MHz)			
Dower Dissinction (mW)	2.70	0.63	1.51			
Power Dissipation (III w)	(@640MHz)	(@320MHz)	(@640MHz)			
Time Resolution(ps)	4.694	10.648	6.655			
Output frequency (MHz)	160MHz	80MHz	160MHz			
Output frequency (MHZ)	~ 640MHz	~ 320MHz	~ 640MHz			

 Table 4.3 Simulation with PVT Variation at the 1.0V Supply

Table 4.4 Simulation with PVT Variation at the 0.5V Supply

Parameter	Post-sim(FF)	Post-sim(SS)	Post-sim(TT)	
Power supply (V)	0.55	0.45	0.5	
Total Cumont (uA)	198.64	52.36	113.8	
Total Current (µA)	(@160MHz)	(@80MHz)	(@160MHz)	
Derver Dissinction (UW)	109.25	23.56	56.90	
Power Dissipation ( $\mu w$ )	(@160MHz)	(@80MHz)	(@160MHz)	
Time Resolution (ps)	16.744	105.48	41.036	
Output frequency (MII)	40MHz	10MHz	40MHz	
Output frequency (MHZ)	~ 320MHz	~ 80MHz	~ 160MHz	

### **4.3 Jitter of Output Frequency Simulation**

We also simulate the jitter performance of the proposed ADPLL. The cycle-to-cycle jitter at 640MHz and 1.0V supply voltage is 6.68ps as shown in Fig. 4.6. Fig. 4.7 shows the period jitter with the same setting is 45.17ps.



Fig. 4.7 Period Jitter at 640MHz, 1.0V Supply Voltage

The cycle-to-cycle jitter at 80MHz and 0.5V supply voltage is 51.52ps as shown in Fig. 4.8. Fig. 4.9 shows the period jitter with the same setting is 199.46ps.



Fig. 4.9 Period Jitter at 80MHz, 0.5V Supply Voltage

#### 4.4 Chip Summary and Comparison Table

Process	90nm	CMOS	
Chip Area	0.09	0.09mm <sup>2</sup>	
Lock-in Time	4 cycles		
Supply Voltage	1.0V	0.5V	
Time Resolutions	6.65ps	41.03ps	
Operating Range	160MHz ~ 640MHz	40MHz ~ 160MHz	
Power Consumption	1.51mW	56.90µW	

Table 4.5 Chip summary of Simulation Result

The chip summary is shown in Table 4.4. The chip is implemented in TSMC 90nm CMOS process. The core area is 0.09 mm<sup>2</sup>.

The frequency range of the proposed ADPLL is about 160MHz to 640MHz and 40MHz to 160MHz at 1.0V and 0.5V, respectively. In addition, the power consumption is 1.51mW and  $56.90\mu$ W at 1.0V and 0.5V, respectively.

Table 4.5 lists the comparisons with prior studies. Although the ADPLL [29] can achieve lock in two clock cycles, it still has a large frequency error due to the on-chip variations of the three DCOs and the quantization effects of the frequency counter. The proposed ADPLL has lowest power consumption and a relatively fast lock-in time. As a result, it is suitable for biomedical electronic applications with DVFS scheme.

Parameter	Proposed	[3] JSSC' 12	[12] TCAS-I '10	[23] JSSC '11	[26] JSSC '03	[29] TCAS-II '10	[34] ASSC '12
Process	90nm	130 nm	90 nm	65 nm	65 nm	180 nm	40 nm
Core Area (mm2)	0.09	0.07	0.27	0.07	1.17	0.075	0.037
Category	ADPLL	Analog PLL	Analog PLL	ADPLL	ADPLL	ADPLL	ADPLL
Supply Voltage	0.5V/1.0V	0.5V	0.5V	1.0V	5V	1.8V	0.5V
Input Frequency (MHz)	5	1.832	280	0.036~12.5	0.011~0.339	0.22~8	1~10
Output Frequency (MHz)	40~160 @0.5V 160~640 @1.0V	400 ~ 433	160 ~ 2500	2~ 700	0.045~ 61.3	222.6 ~ 445.8	10 ~ 100
Time Resolution (ps)	41@0.5V 6.6@1.0V	300 MHz/V	1200 MHz/V	17.5	170	8.8	351
Multiplication Factor	2~128	32	8	16~5200	4 ~ 1022	45~128	100
Power Consumption	56.90μW @(0.5V,160MHz) 1.51mW @(1.0V,640MHz)	440μW @433MHz	1157µW @2.5GHz	1.81mW @520MHz	N/A	14.5mW @446MHz	45.5μW @0.5V
Lock-In Time	4 cycles	N/A	N/A	<75 cycles	7cycles	2 cycles	<42 cycles

Table 4.6 Comparison Table of Simulation Result

### **4.5 Test Chip Measurement Results**



Fig. 4.10 Microphotograph of the ADPLL

Fig. 4.10 shows the microphotograph of one ADPLL test chip. This test chip is fabricated on a standard performance 90nm CMOS 1P9M process. The core area of this chip is 250µm x 250µm. About the chip floorplan and I/O plan of the proposed ADPLL, there are 18 I/O PADs and 14 power PADs. The test chip is composed of the proposed ADPLL block and the testing block. The proposed ADPLL block contains a frequency finder, an ADPLL Controller, a DLF, a cyclic TDC-embedded DCO, a PFD, and a frequency divider. The testing block contains the test divider. Fig. 4.11(a) shows the measured peak-to-peak jitter histogram of the output clock. In this figure, the reference clock is 9.375MHz, and multiplication factor is equal to 64 at 1.0V supply voltage. The target output frequency is 600MHz. Due to the speed limitation of the I/O pad, the signal is divided to 2 by a output divider before output to the I/O pad. The peak-to-peak jitter is 102ps, and the rms jitter is 13.7ps. Fig. 4.11(b) shows the measured frequency of output clock at 600MHz with the same setting.



(a)



Fig. 4.11 (a) Measured Jitter Histogram of Output Clock at 1.0V, 600MHz, the output

is divided by 2

(b) Measured Frequency of Output Clock at 1.0V, 600MHz, the output is

divided by 2

Fig. 4.12(a) shows the measured peak-to-peak jitter histogram of the output clock at 60MHz. In this figure, the reference clock is 7.5MHz, and multiplication factor is equal to 8 at 1.0V supply voltage. The target output frequency is 60MHz. The peak-to-peak jitter is 110ps, and the rms jitter is 15.6ps. Fig. 4.12(b) shows the measured frequency of output clock at 60MHz with the same setting.



(a)


Fig. 4.12 (a) Measured Jitter Histogram of Output Clock at 1.0V, 60MHz

(b) Measured Frequency of Output Clock at 1.0V, 60MHz

Fig. 4.13(a) shows the measured peak-to-peak jitter histogram of the output clock at 120MHz. In this figure, the reference clock is 7.5MHz, and multiplication factor is equal to 16 at 0.52V supply voltage. The target output frequency is 120MHz. The peak-to-peak jitter is 155ps, and the rms jitter is 26.8ps. Fig. 4.13 (b) shows the measured frequency of output clock at 120MHzwith the same setting.



(a)



Fig. 4.13 (a) Measured Jitter Histogram of Output Clock at 0.52V, 120MHz

(b) Measured Frequency of Output Clock at 0.52V, 120MHz

Fig. 4.14(a) shows the measured peak-to-peak jitter histogram of the output clock at 30MHz. In this figure, the reference clock is 7.5MHz, and multiplication factor is equal to 4 at 0.52V supply voltage. The target output frequency is 30MHz. The peak-to-peak jitter is 220ps, and the rms jitter is 28.1ps. Fig. 4.14(b) shows the measured frequency of output clock at 30MHz with the same setting.



(a)



Fig. 4.14 (a) Measured Jitter Histogram of Output Clock at 0.52V, 30MHz

(b) Measured Frequency of Output Clock at 0.52V, 30MHz

Process	90nm CMOS					
Chip Area	0.09mm <sup>2</sup>					
Lock-in Time	4 cycles					
Supply Voltage	1.0V	0.52V				
Time Resolutions	7.3ps	20.9ps				
Operating Range	60MHz ~ 600MHz	30MHz ~ 120MHz				
Power Consumption	0.92mW	37.13µW				

Table 4.7 Chip Summary of Measurement Result

The test chip summary is shown in Table 4.7. The chip is implemented in TSMC 90nm CMOS process. The core area is  $0.09 \text{ mm}^2$ . The frequency range of the proposed ADPLL is about 60MHz to 600MHz and 30MHz to 120MHz at 1.0 V and 0.52V, respectively. In addition, the power consumption is 0.92mW and 37.13 $\mu$ W at 1.0V and 0.52V, respectively.

Table 4.8 shows the comparison table of measurement result. At the low voltage, although the analog CP-based PLLs [3], [12] have a higher output frequency range and lower jitter at a low voltage, the leakage power problem become serious in the 90nm process, it is hard to use the FBB technology for the charge-pump based PLL/DLL working at the low supply voltage. Even if the circuit can be realized, it needs to implement by the triple-well process in the [3], and increase the cost of chip fabrication. Moreover, the charge-pump is difficult to ignore the static power consumption in advanced process. In addition, the control voltage ripple making frequency migration problem will become more serious.

Parameter	Proposed	[3] JSSC' 12	[12] TCAS-I '10	[23] JSSC '11	[26] JSSC '03	[29] TCAS-II '10	[34] ASSC '12
Process	90nm	130 nm	90 nm	65 nm	65 nm	180 nm	40 nm
Core Area (mm2)	0.09	0.07	0.27	0.07	1.17	0.075	0.037
Category	ADPLL	Analog PLL	Analog PLL	ADPLL	ADPLL	ADPLL	ADPLL
Supply Voltage	0.52V/1.0V	0.5V	0.5V	1.0V	5V	1.8V	0.5V
Input Frequency (MHz)	5~20	1.832	280	0.036~12.5	0.011~0.339	0.22~8	1~10
Output Frequency (MHz)	30~120 @0.52V 60~600 @1.0V	400 ~ 433	160 ~ 2500	2~ 700	0.045~ 61.3	222.6 ~ 445.8	10 ~ 100
Jitter (peak-to-peak)	110ps @(0.5V,120MHz) 102ps @(1.0V,600MHz)	49ps @433MHz	18ps @2.24GHz	391ps @52MHz	2ns @38MHz	70ps @446MHz	300ps @100MHz
Time Resolution (ps)	20.9@0.5V 7.3@1.0V	300 MHz/V	1200 MHz/V	17.5	170	8.8	351
Multiplication Factor	2~128	32	8	16~5200	4 ~ 1022	45~128	100
Power Consumption	37.13µW @(0.5V,160MHz) 0.92mW @(1.0V,600MHz)	440μW @433MHz	1157μW @2.5GHz	1.81mW @520MHz	N/A	14.5mW @446MHz	45.5μW @(0.5V,100M Hz)
Lock-In Time	4 cycles	N/A	N/A	<75 cycles	7cycles	2 cycles	<42 cycles

Table 4.8 Comparison Table of Measurement Result

# **Chapter 5**

## **Conclusion and Future Works**

#### **5.1 Conclusion**

In this thesis, a fast lock-in all-digital phase locked loop circuit for supporting DVFS with biomedical electronic applications is proposed.

The proposed frequency estimation algorithm can use the period ratio information calculated from the cyclic TDC to compute the target DCO control code in four clock cycles

With the proposed frequency estimation algorithm, it can use the period ratio parameters obtained by the cyclic TDC to compute the target DCO control code in four clock cycles. The proposed ADPLL can achieve lock in 4 cycles either at 1.0V or 0.52V, and thus the power consumption can be reduced effectively. The power consumption is 0.92mW and 56.90µW at 1.0V and 0.52V, respectively.

In addition, the proposed interpolator-based fine-tuning architecture can easily solve the DCO non-monotonic response problem either at 1.0V or 0.52V, and further enhance the accuracy of the proposed frequency estimation algorithm. The output frequency range of the proposed ADPLL is about 60MHz to 600MHz and 30MHz to 120MHz at 1.0V and 0.5V, respectively.

The test chip is fabricated in TSMC 90nm process with pulse latch DFFs. It can help in TDC achieve higher accuracy period ratio parameter calculation at a low supply voltage. Besides, it improves the performance of PFD and divider at the low supply voltage.

## **5.2 Future Works**

In this thesis, there are some drawbacks in our ADPLL. The first problem is the jitter performance. In the simulation result in Section 4.3, the period jitter and cycle-to-cycle jitter performs unfavorable results. One possible reason is the mechanism of DLF with large reference clock jitter. Therefore, if we can modify our DLF architecture, the jitter performance will be improved.

There is another issue about the frequency range at a low supply voltage. In most analog PLLs / DLLs, the frequency range is over than 1 GHz. However, our ADPLL cannot achieve such frequency range, because the intrinsic delay of delay cell is increased at the low supply voltage, and that limits the highest frequency of the DCO.

If we can improve the delay cell, then the proposed ADPLL will achieve more contribution at the low supply voltage.

### References

- [1] Wei-Hao Sung, Jui-Yuan Yu, and Chen-Yi Lee, "A robust frequency tracking loop for energy-efficient crystalless WBAN systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 10, pp. 637-641, Oct. 2011.
- [2] Maja Vidojkovic, Yao-Hong Liu, Xiongchuan Huang, Koji Imamura, Guido Dolmans, and Harmkede Groot, "A fully integrated 1.7-2.5GHz 1mW fractional-n PLL for WBAN and WSN applications," *Radio Frequency Integrated Circuits Symposium, 2012. RFIC 2012. IEEE*, pp. 185–188, Jun. 2012.
- [3] Wu-Hsin Chen, Wing-Fai Loke, and Byunghoo Jung, "A 0.5-V, 440-μW frequency synthesizer for implantable medical devices, " *IEEE Journal of Solid-State Circuits*, vol. 47, no. 8, pp. 1896-1907, Aug. 2012.
- [4] Jui-Yuan Yu, Chien-Ying Yu, Shang-Bin Huang, Tsan-Wen Chen, Juinn-Ting Chen, Kuan-Ling Kuo, and Chen-Yi Lee, "A 0.5V 4.85Mbps dual-mode baseband transceiver with extended frequency calibration for biotelemetry applications," in *Proc. IEEE Asian Solid-State Circuits Conference*, Nov. 2008, pp. 293-296.
- [5] Farzad Inanlou, Mehdi Kiani, and Maysam Ghovanloo, "A 10.2 Mbps pulse harmonic modulation based transceiver for implantable medical devices," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1296–1306, Jun. 2011.
- [6] Kenji Shiba, Akira Morimasa, and Harutoyo Hirano, "Design and development of low-loss transformer for powering small implantable medical devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, no. 2, pp. 77-85, Apr. 2010.

- [7] Li Fang-Wei, and Han Li, "Dynamic voltage and frequency scaling for power saving in TD-SCDMA," in Proceeding of International Conference on Educational and Information Technology (ICEIT), 2010, pp. 33-37.
- [8] Vivienne Sze and Anantha P. Chandrakasan, "A 0.4-V UWB baseband processor," in Proceedings of ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), Aug. 2007, pp. 262-267.
- [9] Myeong-Eun Hwang, Arijit Raychowdhury, Keejong Kim, and Kaushik Roy,
   "A 85mV 40nW process-tolerant subthreshold 8x8 FIR filter in 130nm technology," *in Digest of Technical Papers, Symposium on VLSI Circuits*, Jun. 2007, pp. 154-155.
- [10] Alice Wang and Anantha P. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310-319, Jan. 2005.
- [11] Patrick P. Mercier, Manish Bhardwaj, Denis C. Daly, and Anantha P. Chandrakasan, "A low-voltage energy-sampling IR-UWB digital baseband employing quadratic correlation," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 6, pp. 1209-1219, Jun. 2010.
- [12] Kuo-Hsing Cheng, Yu-Chang Tsai, Yu-Lung Lo, and Jing-Shiuan Huang, "A
   0.5-V 0.4-2.24 GHz inductorless phase-locked loop in a system-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 5, pp. 849-854, May. 2011.
- [13] Tianwang Li, Bo Ye, and Jinguang Jiang, "0.5V 1.3GHz voltage controlled ring oscillator," *in Proceedings of IEEE International Conference on ASICON*, Oct. 2009, pp. 1181-1184.
- [14] Hsieh-Hung Hsieh, Chung-Ting Lu, and Liang-Hung Lu, "A 0.5-V 1.9-GHz low-power phase-locked loop in 0.18μm CMOS," in Digest of Technical Papers, - 71 -

Symposium on VLSI Circuits, Jun. 2007, pp. 164-165.

- [15] Chung-Ting Lu, Hsieh-Hung, and Liang-Hung Lu, "A low-power quadrature VCO and its application to a 0.6-V 2.4-GHz PLL," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 793-802, Apr. 2010.
- [16] Ching-Yuan Yang, Chih-Hsiang Chang, Jun-Hong Weng, and Hsin-Ming Wu,
  "A 0.5V/0.8-V 9-GHz frequency synthesizer with doubling generation in
  0.13µm CMOS," *IEEE Transactions on Circuits and Systems II: Express Briefs*,
  vol. 58, no. 2, pp. 65-69, Feb. 2011.
- [17] Chih-Hsiang Chang, and Ching-Yuan Yang, "A low-voltage 9GHz 0.13μm CMOS frequency synthesizer with a fractional phase-rotating and frequency doubling topology," *in Digest of Technical Papers, Symposium on VLSI Circuits*, Jun. 2009, pp. 192-193.
- [18] Shih-An Yu, and Peter Kinget, "A 0.65-V 2.5-GHz fractional-N synthesizer with two-point 2-Mb/s GFSK data modulation," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 9, pp. 2411-2425, Sep. 2009.
- [19] Jose A. Tierno, Alexander V. Rylyakov, and Daniel J. Friedman, "A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65nm SOI," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 42-51, Jan. 2008.
- [20] Tae-Hyoung Kim, John Keane, Hanyong Eom, and Chris H. Kim, "Utilizing reverse short-channel effect for optimal subthreshold circuit design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 7, pp. 821-829, Jul. 2007.
- [21] Ching-Che Chung, and Chen-Yi Lee, "An all-digital phase-locked loop for high-speed clock generation," *IEEE Journal of Solid-State Circuits*, vol.38, no. 2, pp. 347-351, Feb. 2003.

- [22] Hsuan-Jung Hsu, and Shi-Yu Huang, "A low-jitter ADPLL via a suppressive digital filter and an interpolation-based locking scheme," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 1, pp. 165-170, Jan. 2011.
- [23] Ching-Che Chung, and Chiun-Yao Ko, "A fast phase tracking ADPLL for video pixel clock generation in 65nm CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 46, no.10, pp. 2300-2311, Oct. 2011.
- [24] Chao-Ching Hung, and Shen-Iuan Liu, "A 40-GHz Fast-Locked All-Digital Phase-Locked Loop Using a Modified Bang-Bang Algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 6, pp. 321-325, Jun. 2011.
- [25] Guang-Kaai Dehng, June-Ming Hsu, Ching-Yuan Yang, and Shen-Iuan Liu, "Clock-deskew buffer using a SAR-controlled delay-locked loop," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1128-1136, Aug. 2000.
- [26] Takamoto Watanabe, and Shigenori Yamauchi, "An all-digital PLL for frequency multiplication by 4 to 1022 with seven-cycle lock time," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 198-204, Feb. 2003.
- [27] Liming Xiu, and Zhihong You, "A "flying-adder" architecture of frequency and phase synthesis with scalability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 1, pp. 165-170, Jan. 2011.
- [28] Gang-Neng Sung, Szu-Chia Liao, Jian-Ming Huang, Yu-Cheng Lu, and Chua-Chin Wang, "All-digital frequency synthesizer using a flying adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 8, pp. 597-601, Aug. 2010.
- [29] Chia-Tsun Wu, Wen-Chung Shen, Wei Wang, and An-Yeu Wu, "A two-cycle lock-in time ADPLL design based on a frequency estimation algorithm," *IEEE* - 73 -

Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 6, pp. 430-434, Jun. 2010.

- [30] Seungsoo Kim, and Hyunchol Shin, "An E-TSPC divide-by-2 circuit with forward body biasing in 0.25µm CMOS," *IEEE Microwave and Wireless Components Letters*, vol. 19, no. 10, pp. 656-658, Oct. 2009.
- [31] Chulwoo Kim, and Sung-Mo (Steve) Kang, "A low-swing clock double-edge triggered flip-flop," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 5, pp. 648-652, May 2002.
- [32] Duo Sheng, and Jhih-Ci Lan, "A monotonic and low-power digitally controlled oscillator with portability for SoC applications," in Proceedings of IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Aug. 2011.
- [33] Ching-Che Chung, and Wei-Cheng Dai, "A referenceless alldigital fast frequency acquisition full-rate CDR circuit for USB 2.0 in 65nm CMOS technology," in Proceedings of International Symposium on VLSI Design, Automation, and Test (VLSI-DAT), Apr. 2011, pp. 217-220.
- [34] Yasuyuki Hiraku, Isamu Hayashi, Hayun Chung, Tadahiro Kuroda, and Hiroki Ishikuro, "A 0.5V 10MHz-to-100MHz 0.47µW/MHz Power Scalable AD-PLL in 40nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conference*, Nov. 2012, pp. 33-36.