

國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

一個具有超寬輸出頻率範圍的 CMOS 鎖相迴路
A CMOS Phase-Lock Loop with an Ultra-Wide Output
Frequency Range

張文豪

Wen-Hao Chang

指導教授 呂良鴻 博士

Advisor: Liang-Hung Lu, Ph.D.

中華民國 113 年 05 月

May 2024





國立臺灣大學碩士學位論文

口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

一個具有超寬輸出頻率範圍的 CMOS 鎖相迴路

A CMOS Phase-Locked Loop

with an Ultra-Wide Output Frequency Range

本論文係張文豪(R09943130)在國立臺灣大學電子工程學研究所完成之碩士學位論文，於民國 113 年 5 月 2 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Electronics Engineering on 2, May, 2024 have examined a Master's Thesis entitled above presented by Wen-Hao Chang (R09943130) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

呂良鴻

林彥賢

蔡宗亨

(指導教授 Advisor)

系(所、學位學程)主管 Director:

江介宏



謝辭



一路以來，謝謝我的爸媽，讓我沒有經濟上的困難，對我學業上的壓力有著極大的包容，還有每次通勤時的媽寶專車。謝謝阿公阿嬤，在我借住的這段期間，對我無限的溺愛，還有無限的水煮蛋。謝謝張愷苾，雖然我們很少聯絡，但每次有難以告人的煩惱的時候，你都會回我訊息，擔任我的小間諜。謝謝楊雅婷，你一直是我的目標，你的聲音就能給我力量，雖然我的笑話一直被你嫌棄，但一定是沒有現場講的關係，等我們近距離之後再讓你好好見識。

在研究外，謝謝我所有的朋友們，因為有你們，我的生活才沒那麼無趣。陳祈瑋、陳靖杰、陳姿妤、鄭伊玟、陳孟潔，跟你們一起玩總讓我很放鬆，你們讓我也開始嚮往好朋友們住在同一棟大樓的生活，只好交給你們努力賺錢起厝幫我圓夢了。還有法文課認識的朋友們，Vivian、Thomas、Morgane、Yupei、Kylie、Max、Anne，我沒想到上個課還能交到好朋友，希望我們以後還能維繫感情，還能一起玩劇本殺和去 bla bla night。

在研究上，首先要感謝呂良鴻老師的指導，給了我論文题目的方向，對學生的態度也總是十分友善而有底線。另外我還想特別感謝呂老師給我擔任電子學統籌助教的機會，帶領讀書會的經驗讓我感受到了作為一名指導者，幫助學生理解課程觀念時的成就感，滿足了我想當老師的志願。還有實驗室的各位，謝謝紹哲，每次都是你陪我留在實驗室，還有各方各面不藏私的幫助。謝謝芳瑜，曾在半夜三點跑來實驗室幫助我成功下線。謝謝懷元，換來你的位子後終於畢業了。謝謝博堯、鵬宇，陪我去澎湖，你們是我永遠的兄弟。其他暫時還在實驗室的大家，因為你們都吵著要被我寫在謝辭裡，所以我就一併寫在這了：若穎、亦捷、登榮、紹歲、亮辰、成哲、政頡、弘偉、郁翔、弈勳、博淵、柏言。謝謝你們的陪伴，讓實驗室經常歡聲笑語，讓我流連忘返。祝各位將來扶搖直上九萬里，當然，飛黃騰達之時，切記，勿忘我。

2024.06.02 張文豪





摘要

在當今的多頻多模通信系統中，一個體積小巧、靈活性高且具有高頻譜純度的時脈產生器是不可或缺的。由於其穩定性高、佔用空間小，第一類鎖相迴路(Type-I PLL) 近來已成為許多研究的焦點。然而，第一類鎖相迴路的鎖定範圍受到限制，這不僅使得其在實際應用中無法確保在製程、電壓、溫度變異下仍能成功鎖定，也限制了其作為寬調變範圍(tuning range)時脈產生器的應用。

本論文提出了一個頻帶選擇機制並設計了相應的頻帶選擇迴路。該迴路防止了第一類鎖相迴路脫鎖，並克服了鎖定範圍和參考突波之間的取捨。藉由頻帶選擇迴路的輔助，本論文實現了一個寬調變範圍的第一類鎖相迴路。該晶片採用TSMC 180-nm CMOS製程實現，核心電路面積為 0.12 mm^2 ，操作電壓為1.8 V，輸出頻率範圍為840 - 2240 MHz，提供了超過90%的調變範圍。當輸入參考頻率為10 MHz，輸出頻率為2240 MHz時，在1-MHz頻率偏移的地方，相位雜訊為-91.7 dBc/Hz，參考突波為-50 dBc。在論文的最後，對該晶片進行了優化，計算出了最佳化的參數設計及系統頻率規劃。

此外，本文提出的頻帶選擇迴路易於適應不同的製程，並有望在未來與基於單元(cell-based)的設計流程結合，以簡化設計複雜性。

關鍵字：鎖相迴路、第一類、鎖定範圍、寬調變範圍





Abstract

In today's multi-band and multi-mode communication systems, a compact, flexible, and high-spectral purity clock generator is indispensable. Due to its high stability and small footprint, the type-I PLL has recently become the focus of much research. However, the acquisition range of Type-I PLLs is limited. This not only prevents them from reliably locking under process, voltage, and temperature variations in real-world applications but also restricts their use as wide-tuning range clock generators.

This paper proposes a band-selecting mechanism and designs a corresponding band-selecting loop. The loop prevents the type-I PLL from unlocking and overcomes the trade-off between the acquisition range and the reference spur. With the assistance of the band-selecting loop, this paper realizes a wide tuning range type-I PLL. The chip is implemented in a TSMC 180-nm CMOS process, with a core circuit area of 0.12 mm², operating at 1.8 V. The output frequency range is 840 - 2240 MHz, providing over 90% of the tuning range. At an input reference frequency of 10 MHz and an output frequency of 2240 MHz, the phase noise at 1 MHz frequency offset is -91.7 dBc/Hz, and the reference spur is -50 dBc. In the final section, the optimal parameter design and system frequency planning are calculated.

Furthermore, the band-selecting loop proposed in this paper is easily adaptable to different processes and is expected to be combined with cell-based design flow in the

future to simplify the design complexity.

Keywords: Phase-Locked Loop, Type-I, Acquisition Range, Wide-Tuning Range





Contents

Approval	i
Acknowledgment	iii
Chinese Abstract	v
Abstract	vii
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Overview	3
2 Introduction to Type-I PLL	5
2.1 Fundamentals of the Phase-Locked Loop	5
2.1.1 Voltage-Controlled Oscillators	5
2.1.2 Phase Detectors	8
2.1.3 Steady State Behavior	13
2.1.4 Dynamic Behavior	15
2.2 Transfer Function of Type-I PLL	16
2.3 Limitation of Acquisition Range in Type-I PLL	20
3 Proposed PLL	25
3.1 Overview	25
3.2 Overall Architecture	25

3.3	Proposed Band-Selecting Loop	30
3.3.1	Band-Selecting Mechanism	30
3.3.2	Clock Generator	35
3.3.3	Activation of Band-Selecting Loop	40
3.3.4	The Whole Structure of Band-Selecting Loop	41
3.4	Circuit Implementation	43
3.4.1	Phase Detector and Master-Slave Sampling Filter	44
3.4.2	Multi-Band Voltage Control Oscillator	47
3.4.3	Multi-Modulus Divider	50
3.4.4	Band-Selecting Loop	58
3.5	Summary	60
4	Experimental Results	61
4.1	Chip Photo	61
4.2	Measurement Setup	62
4.3	Experiment Result	63
4.4	Comparison	68
4.5	Summary	68
5	Modified Design	71
5.1	Work Review	71
5.2	Adjusted Block	74
5.3	Phase Noise Consideration	75
5.4	Simulation Result	76
5.5	Summary	80
6	Conclusion	81
	Reference	83

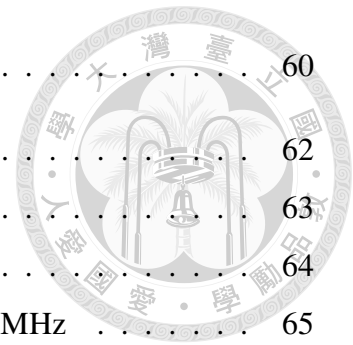


List of Figures

2.1	VCO tuning characteristic	6
2.2	PD sensing two periodic signals	8
2.3	PD characteristic	8
2.4	Waveform of an XOR	9
2.5	Characteristic of XOR	10
2.6	PD may lose its function when the duty cycle isn't 50%	10
2.7	(a) J-K flip-flop (b) Truth table (c) Waveform in steady state	11
2.8	Characteristic of J-K flip-flop	12
2.9	J-K flip-flop produces the same output for harmonic V_2 and V_2'	13
2.10	Block diagram of a simple PLL	13
2.11	Block diagram of a simple PLL with modulus N	14
2.12	Apply a frequency step at the input of a simple PLL	15
2.13	Apply a phase step at the input of a simple PLL	16
2.14	Linear model of a basic PLL	17
2.15	Bode plot of a basic type-I PLL	18
2.16	Cycle slipping	20
2.17	Open-loop PLL and spectrum of V_{PD} and V_{ctrl}	21
2.18	The locking progress	22
2.19	The PLL fails to lock if initial $ \omega_{out} - \omega_{in} $ is too large	23
2.20	Reference spur versus bandwidth for type-I PLL [1]	24
3.1	Unlocking phenomenon when switching divisors	26
3.2	Proposed approach to avoid the limitations of acquisition range	27
3.3	The conventional method of frequency band controlling	28
3.4	The proposed band-selecting process	29

3.5	The block diagram of the proposed wide frequency range Type-I PLL . . .	30
3.6	The FSM of the proposed band-selecting mechanism	31
3.7	The relation between VCO output and reference signal	32
3.8	Use a counter to get the ratio of f_{out} and f_{REF}	32
3.9	Use two comparators to implement state 2	34
3.10	Update control bits by a bi-directional shift register	35
3.11	Sequence of each state in proposed band-selecting mechanism	36
3.12	Three clock signals for respective state	36
3.13	Realize the clock generator by a synchronous counter	37
3.14	schematic of D flip-flop	37
3.15	The complete timing diagram of clock generator	38
3.16	Reset signal in clock generator	39
3.17	Implementation of state 0	40
3.18	The full picture of the proposed BSL	42
3.19	Transient simulation result of BSL	43
3.20	A type-I PLL using a master-slave sampling filter	44
3.21	(a) Nonoverlap generator and S2D (b) Waveform of $\phi_{1,2}$	45
3.22	Waveform in steady state	47
3.23	Current starved ring VCO	48
3.24	Current starved ring VCO	49
3.25	Phase noise of free-run VCO	49
3.26	Pulse swallow divider	51
3.27	Divider structure of this work	52
3.28	Divide-by-two circuit	52
3.29	Dual-modulus $\div 4/5$ circuit	53
3.30	Waveform of the dual-modulus prescaler	54
3.31	Programmable program counter	55
3.32	Waveview of program counter when $P_{in} = 5$	55
3.33	Programmable swallow counter	56
3.34	Waveview of swallow counter when $S_{in} = 2$	56
3.35	Transient simulation of pulse swallow divider	57
3.36	Decoder when $P_{in} = 5$ and $S_{in} = 1$	59

3.37	Connection between VCO and the main loop	60
4.1	Chip photo	62
4.2	Measurement setup	63
4.3	Transient response simulation of the system	64
4.4	Measured spectrum of the proposed type-I PLL at 840 MHz	65
4.5	Measured spectrum of the proposed type-I PLL at 2240 MHz	65
4.6	Measured spectrum when the PLL is unlocked	66
4.7	Measured phase noise of the proposed type-I PLL at 840 MHz	67
4.8	Measured phase noise of the proposed type-I PLL at 2240 MHz	67
5.1	The arrangement of BSL in layout	72
5.2	The adjusted divider is composed of two $\div 2$ circuit and one program counter	74
5.3	All of PLL sub-blocks noise contribution	75
5.4	Layout of the improved PLL	76
5.5	Transient simulation result of the improved system	77
5.6	Output spectrum when $f_{out} = 2240$ MHz	78
5.7	Phase noise when $f_{out} = 800$ MHz	79
5.8	Phase noise when $f_{out} = 2240$ MHz	79







List of Tables

4.1 Comparison table	69
--------------------------------	----





Chapter 1

Introduction

1.1 Motivation

From computer processors to communication systems, reliable clock generators are widely employed in various electronic systems to generate stable timing signals. In wireless communication systems, the local oscillator (LO) is indispensable. To avoid close talk or reduce signal-to-noise ratio (SNR), a LO with high stability, and low phase noise is required. Moreover, the capability to generate different output frequencies is also essential for a multi-channel wireless communication system. To meet these requirements, the Phase-Locked Loop (PLL) is deemed the most suitable technology [2]. One of the earliest integrated circuit (IC)-based PLLs was introduced in 1969 [3]. Since then, the design of PLL-based frequency synthesizers has garnered significant attention. Countless research focused on optimizing the performance, power consumption, and area (PPA) metrics for PLLs [4].

The typical type-II charge-pump (CP) PLL is the most common architecture. Thanks to the well-known linearized system model, the behavior of such architecture can be estimated

accurately. Additionally, detailed design processes are documented [5][6]. However, the type-II PLL suffers from several limitations [7][8][9][10], including inherent instability caused by the two poles at the origin from CP and the voltage-controlled oscillator (VCO), as well as the restriction imposed by Gardner's limit [11], which mandates that the system bandwidth must be at least less than $f_{REF}/10$. These issues necessitate a large loop filter (LF) for a Type-II PLL, often resulting in the LF occupying an area even larger than the inductor in the LC oscillator [12][13].

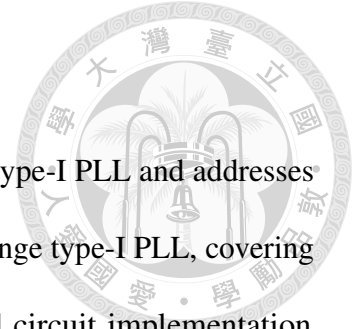
On the other hand, in the architecture of a Type-I PLL, the VCO serves as the only integrator in the loop. Therefore, it guarantees system stability, eliminating the need for a large LF, and making it a relatively compact structure. Furthermore, type-I PLLs are not subject to Gardner's limit, and theoretically, their bandwidth can extend up to $f_{REF}/2$ [1]. A larger loop bandwidth can effectively suppress the phase noise of the VCO to a greater extent. Consequently, research on type-I PLLs has been garnering increasing attention recently.

However, for type-I PLLs, the limited acquisition range is a significant issue. A narrow acquisition range can cause the loop to fail to lock under PVT variations in practical scenarios. This also limits the Type-I PLL's ability to provide a wide tuning range and flexible application in multiband systems. Although increasing the bandwidth can indeed expand the locking range [14], the reference spur worsens as the bandwidth increases. In other words, there is a trade-off between the acquisition range and the reference spur.

In this work, we introduce a novel mechanism to overcome the limited acquisition range, breaking the trade-off between the acquisition range and the reference spur. With the proposed band-selecting loop, we will present a wide tuning range with multiple modulus type-I PLL.

1.2 Thesis Overview

The thesis is organized as follows: Chapter 2 introduces the type-I PLL and addresses its critical issues. Chapter 3 presents the proposed wide tuning range type-I PLL, covering its architectures, the proposed band-selecting mechanism, and circuit implementation. Chapter 4 displays the experimental results. Chapter 5 discusses areas for improvement in the current chip and presents layout and simulation results. Finally, Chapter 6 provides a brief conclusion.







Chapter 2

Introduction to Type-I PLL

2.1 Fundamentals of the Phase-Locked Loop

In communication systems, an accurate local oscillator (LO) frequency is crucial. Even a minor shift causes significant spillage of high-power interference into a desired channel. Thus, a synthesizer that can set LO frequency precisely is needed. Phase-Locked Loop (PLL) is a structure that minimizes the frequency drift and phase noise. This section will discuss the architecture of a basic PLL. Start from voltage-controlled oscillators (VCO), phase detectors (PD), and then the loop analysis of a simple PLL.

2.1.1 Voltage-Controlled Oscillators

A VCO can be seen as a block with input voltage and output frequency. The ideal relationship between input and output is a straight line, shown in Figure 2.1. The VCO tuning characteristic can be modeled as the equation

$$\omega_{out} = K_{VCO}V_{cont} + \omega_0 \quad (2.1)$$

where K_{VCO} is the gain of VCO and its unit is rad/s/V.

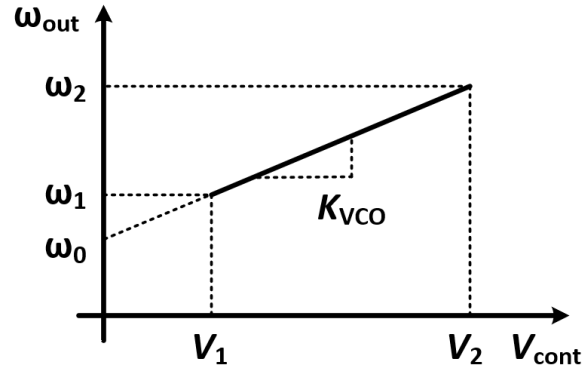


Figure 2.1: VCO tuning characteristic

We can also analyze VCO in the time domain. For the case that the control voltage of VCO is a constant, we could write the equation easily:

$$V_{out} = V_0 \cos(\omega_{out}t) \quad (2.2)$$

$$= V_0 \cos(\omega_0 t + K_{VCO} V_{cont} t) \quad (2.3)$$

But what if V_{cont} is not a constant? Recall that phase is the time integral of frequency, so the time-domain phase relationship is

$$\phi_{out}(t) = \int_0^t \omega_{out}(t) dt \quad (2.4)$$

$$= \omega_0(t) + K_{VCO} \int_0^t V_{cont}(t) dt \quad (2.5)$$

Assuming K_{VCO} remains constant with time, we can get

$$V_{out}(t) = V_0 \cos [\omega_0(t) + K_{VCO} \int_0^t V_{cont}(t) dt] \quad (2.6)$$



Here we define the "excess phase," ϕ_{ex} , as

$$\phi_{ex} = K_{VCO} \int_0^t V_{cont}(t) dt \quad (2.7)$$

In PLLs, we primarily consider the VCO as a Linear Time-Invariant (LTI) system with V_{ctrl} as the input and the excess phase ϕ_{ex} as the output. We will use this excess phase for the following analysis.

Now, we can take Laplace to transform to achieve a transfer function for the VCO in s-domain:

$$\frac{\phi_{out}(s)}{V_{ctrl}(s)} = \frac{K_{VCO}}{s} \quad (2.8)$$

According to the analysis above, we can conclude that (a) we should change the frequency to change the phase of VCO output, which is how a PLL achieves phase-locked. (b) VCO is a dynamic system. The output phase of the VCO cannot be determined straight from the control voltage but depends on the past value of the control voltage. (c) The integrator property implies that the phase of a VCO cannot be changed instantaneously, we have to wait for $K_{VCO} \int V_{cont}(t) dt$ until it reaches the desired value after changing the control voltage.

2.1.2 Phase Detectors

PD can measure the phase difference between two input signals and produce an output voltage. The concept is illustrated in Figure 2.2. The phase difference $\Delta\phi$ between $V_1(t)$ and $V_2(t)$ is detected by PD and then generates an output voltage $V_{out}(t)$.

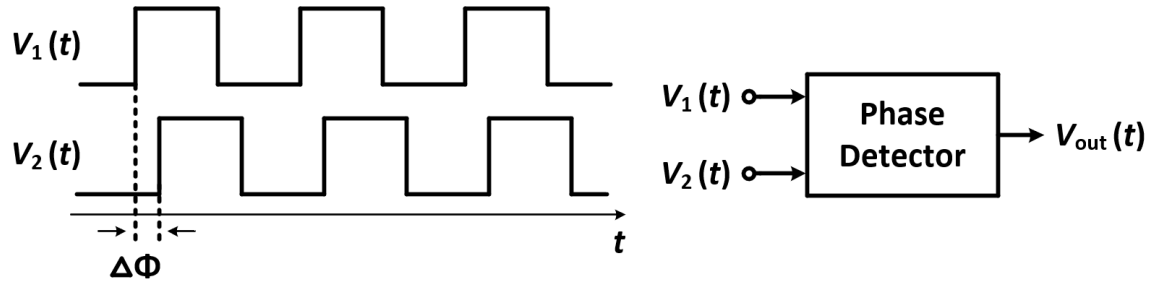


Figure 2.2: PD sensing two periodic signals

The behavior of PD is plotted in Figure 2.3. Ideally, the average output voltage should be proportional to the sensing phase difference. The slope is denoted by K_{PD} , which represents the gain of PD.

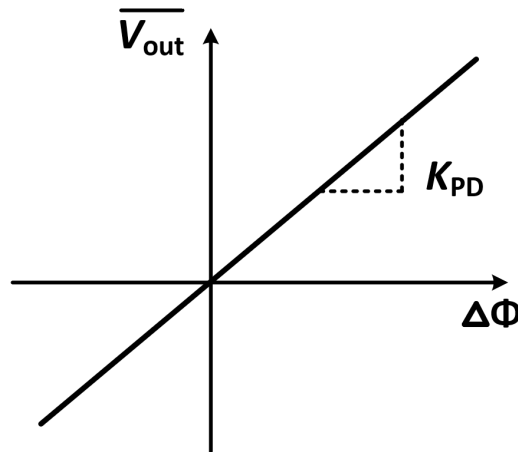


Figure 2.3: PD characteristic

According to the description above, the transfer function of PD can be summarized as a constant value

$$H_{PD}(s) = K_{PD} \quad (V/rad) \quad (2.9)$$

There are various methods to implement a PD. We'll provide a brief introduction to two of them: the Exclusive-or (XOR) gate and the J-K flip-flop.



- XOR Phase Detector

Figure 2.4 shows the operation of an XOR gate. It generates pulses whose width is equal to the input phase difference $\Delta\phi$. Transfer $\Delta\phi$ to the time domain, we can have

$$\Delta t = T_{in} \times \frac{\Delta\phi}{2\pi} \quad (2.10)$$

The average output can be calculated as

$$\overline{V_{out}} = \frac{2\Delta t}{T_{in}} \times V_{DD} = \frac{V_{DD}}{\pi} \times \Delta\phi \quad (2.11)$$

As a result, we can conclude that

$$K_{PD} = \frac{\overline{V_{out}}}{\Delta\phi} = \frac{V_{DD}}{\pi} \quad (2.12)$$

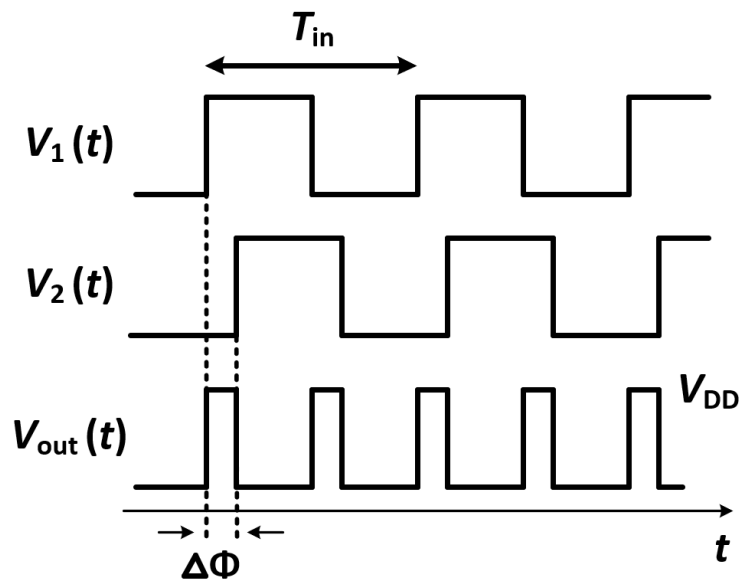


Figure 2.4: Waveform of an XOR

Furthermore, since the pulse width remains consistent for both leading and lagging phase differences, the average output is still linear but in the opposite direction for every interval where $V_2(t)$ leads $V_1(t)$, such as $-\pi$ to 0 or π to 2π . We can observe that the gain $K_{PD} = -V_0/\pi$. The input-output characteristic is illustrated in Figure 2.5.

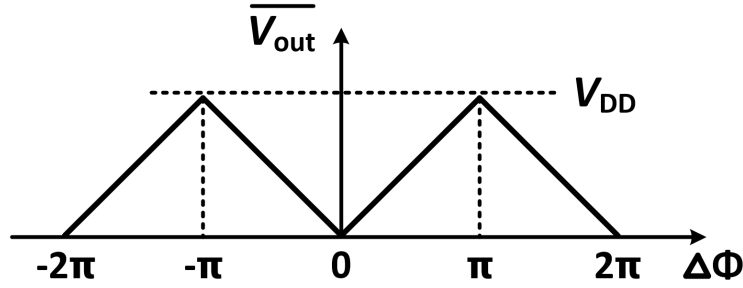


Figure 2.5: Characteristic of XOR

There are some limitations to an XOR-based PD. One is that it is sensitive to the duty cycle of input signals. It requires the input signals with a duty cycle close to 50%. An excessively large or small duty cycle can affect the range of phase differences it can detect. As shown in Figure 2.6, the average outputs are the same, regardless of the input phase difference $\Delta\phi_1$ and $\Delta\phi_2$.

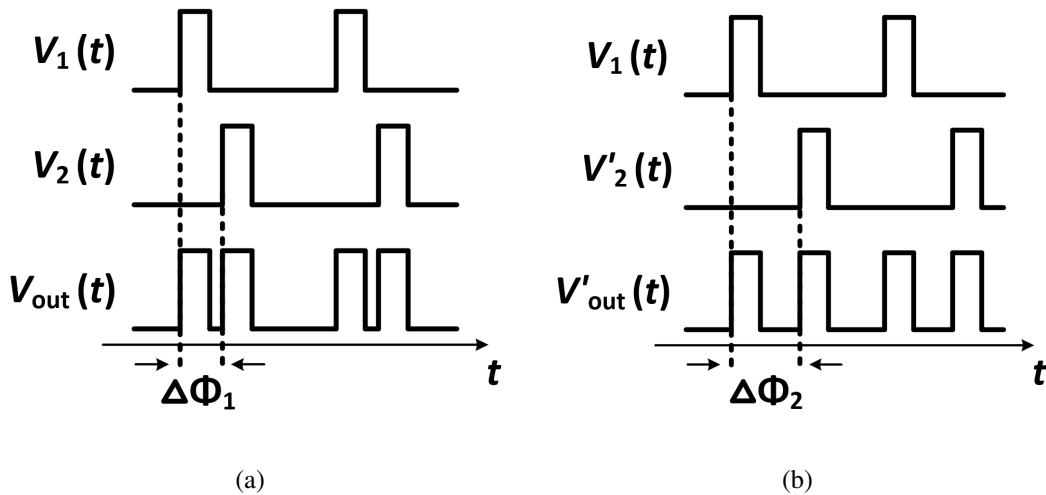


Figure 2.6: PD may lose its function when the duty cycle isn't 50%

Another limitation arises from its two gain stages with opposite polarities. This makes it difficult to determine whether the system is in a positive or negative feedback operation. In a PLL, this can lead to phenomena such as "Cycle Slipping" or "Unlock". We will discuss these two situations later after reviewing the overall PLL architecture.

- J-K Flip-Flop Phase Detector

J-K flip-flop is also a circuit that can generate pulses with a width equal to $\Delta\phi$. The truth table and the waveform of a J-K flip-flop are illustrated in Figure 2.7.

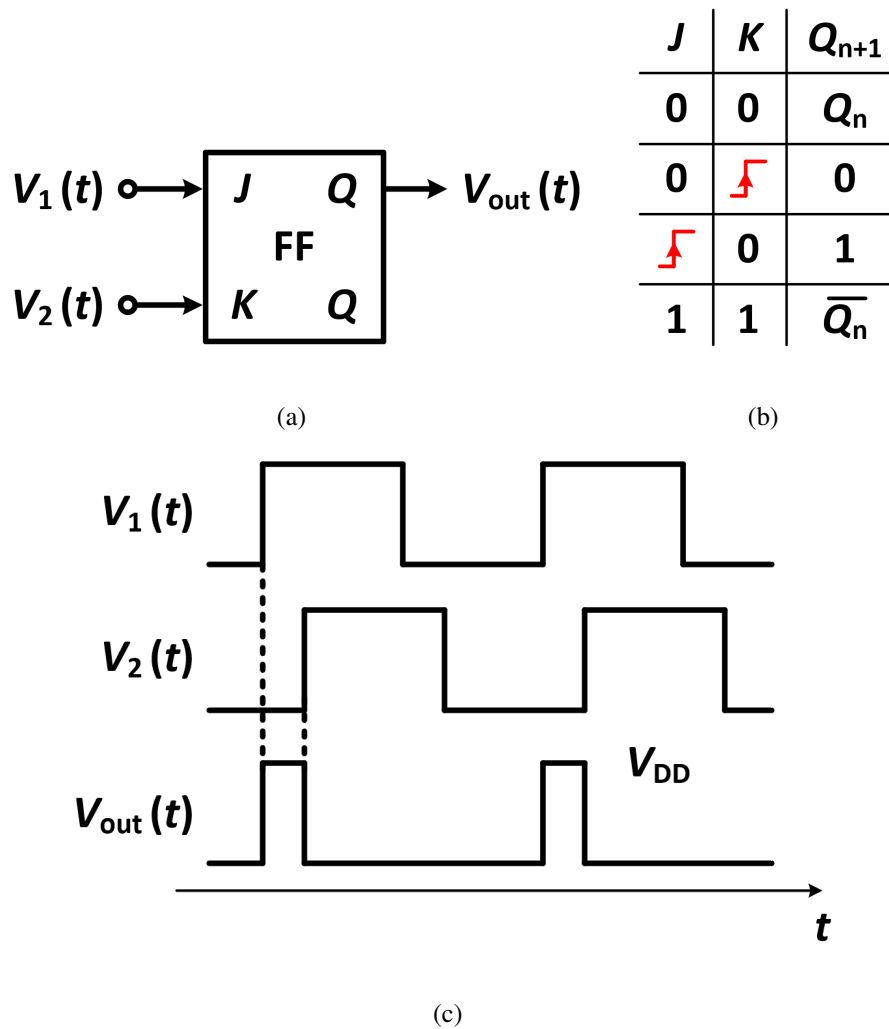


Figure 2.7: (a) J-K flip-flop (b) Truth table (c) Waveform in steady state

The average output of J-K flip-flop is written as

$$\overline{V_{out}} = \frac{\Delta t}{T_{in}} \times V_{DD} = \frac{V_{DD}}{2\pi} \times \Delta\phi \quad (2.13)$$



So the gain of J-K flip-flop PD is calculated

$$K_{PD} = \frac{\overline{V_{out}}}{\Delta\phi} = \frac{V_{DD}}{2\pi} \quad (2.14)$$

The J-K flip-flop exhibits a constant gain K_{PD} over a 2π range. Even though K leads J due to the negative phase difference, it operates similarly to when J leads K , resulting in K_{PD} remaining positive. The input-output characteristic is depicted in Figure 2.8.

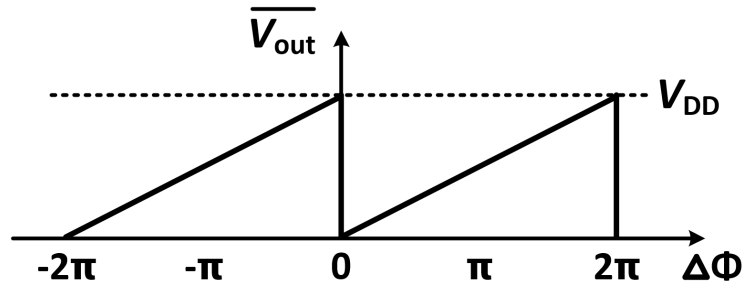


Figure 2.8: Characteristic of J-K flip-flop

Unlike XOR-based PD, J-K flip-flop is not sensitive to the input duty cycle because it is triggered by rising edges. And because its gain is always positive, it does not induce cycle clipping in the PLL application. However, harmonic signals can produce the same output, as shown in Figure 2.9, potentially resulting in locking to harmonics.

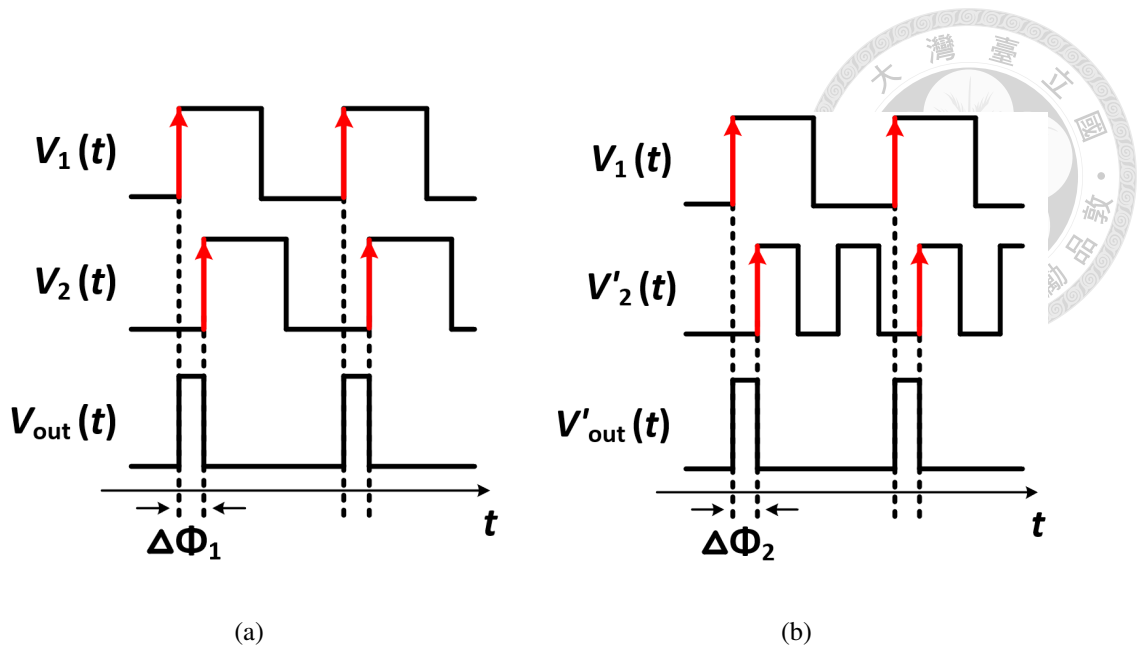


Figure 2.9: J-K flip-flop produces the same output for harmonic V_2 and V'_2

2.1.3 Steady State Behavior

A simple PLL consists of PD, LPF, and VCO. The PD quantifies the disparity between the two phases of the reference signal and the feedback signal. The LPF averages the output of PD to obtain a DC level as V_{ctrl} for the VCO. The VCO varies its output frequency according to V_{ctrl} . Without a frequency divider, the output signal straightly feedbacks to PD. The block diagram is shown in Figure 2.10.

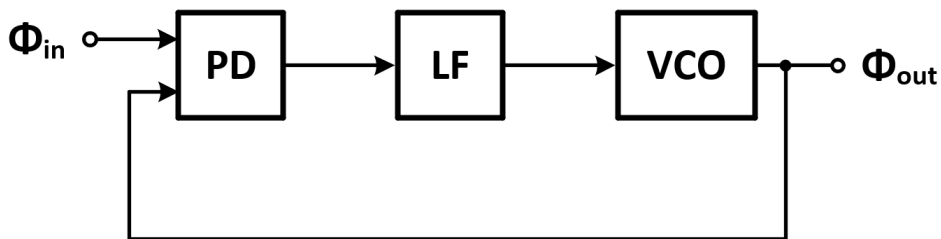


Figure 2.10: Block diagram of a simple PLL

We say the loop is "locked" if $\phi_{out} - \phi_{in}$ is constant. Thus, we can describe the steady

state as

$$\frac{d\phi_{out}}{dt} = \frac{d\phi_{in}}{dt} \quad (2.15)$$

As we mentioned in Section 2.1.1, phase is the time integral of frequency, so the time derivative of the phase is equal to the instant frequency. We have:

$$2\pi f_{out} = 2\pi f_{in} \quad (2.16)$$

This equation demonstrates that when the phases are locked, the output frequency is also "locked," equal to the input frequency.

Now, let's incorporate the frequency divider into consideration. With a divider of modulus N , as illustrated in Figure 2.11, an important property of the PLL is demonstrated: frequency multiplication. At steady-state, the relation between the input frequency and the output frequency is defined as

$$f_{out} = N \times f_{REF} \quad (2.17)$$

The divider is utilized to achieve the desired frequency synthesis. With a variable N , the PLL can generate different output frequencies and configure various channels in the wireless communication system.

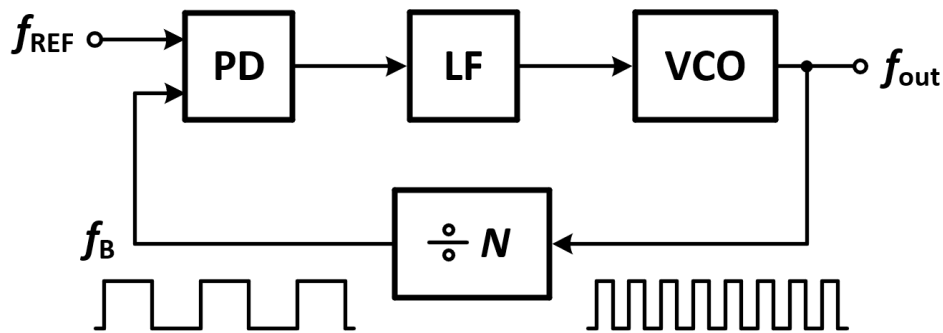


Figure 2.11: Block diagram of a simple PLL with modulus N

2.1.4 Dynamic Behavior

As we study the dynamic behavior of a circuit, we usually apply a step at its input and observe the transient response. For a PLL, the PD recognizes the input signal as either the frequency or the phase. Thus, we can view step response from the perspective of both. For ease of explanation, let's consider a basic PLL without a frequency divider as Figure 2.10.

Let's start with an input frequency step. Initially, assume that both input and output are stable with a frequency of f_0 . As depicted in Figure 2.12, there is a frequency jump from f_0 to $f_0 + \Delta f$ at $t = t_0$. The LPF maintains V_{ctrl} at V_0 at first, so the VCO continues to oscillate at f_0 . Due to the frequency difference Δf between the input and output frequencies, their phase difference gradually increases. This causes the pulses generated by PD to become wider and wider, leading to an increase in V_{ctrl} and consequently f_{out} . This process keeps going until f_{out} becomes $f_1 = f_0 + \Delta f$. PLL finally locks at a new frequency again. Note that the settling behavior of V_{ctrl} is related to the stability of the system, i.e., the damping factor of the loop.

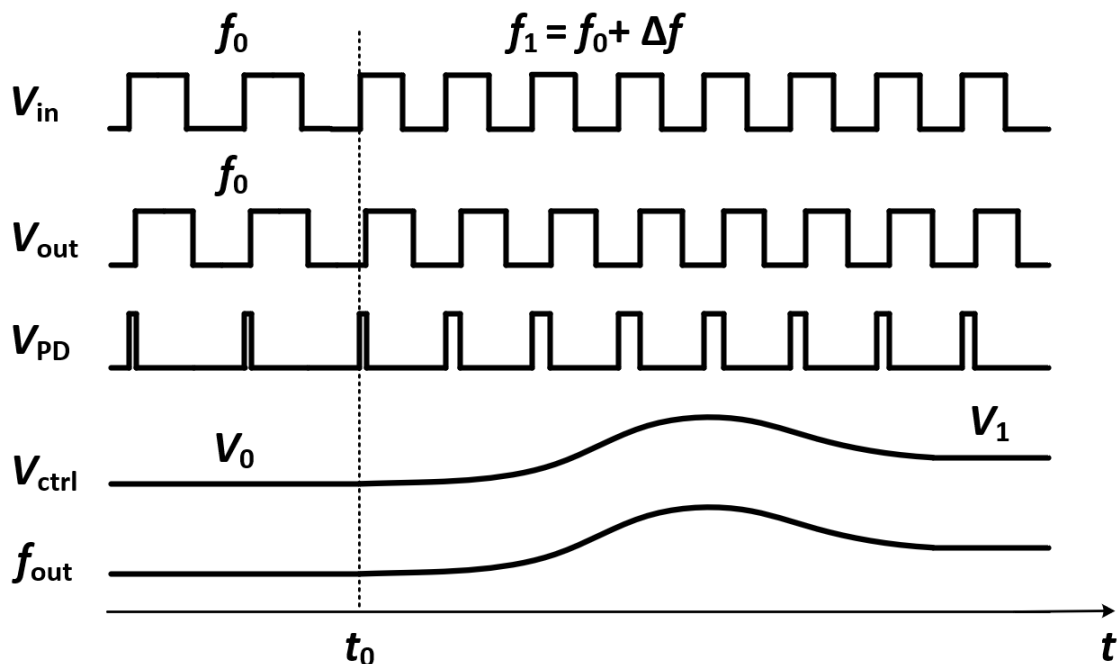


Figure 2.12: Apply a frequency step at the input of a simple PLL

The response of the PLL to an input phase step is illustrated in Figure 2.13. When a phase step is applied at $t = t_0$, it introduces additional phase error, resulting in wider PD output pulses. As V_{ctrl} increases, f_{out} also rises, allowing the VCO to accumulate additional phase. Eventually, the output phase catches up with the input phase, compensating for the added phase step, and the final frequency matches the initial one.

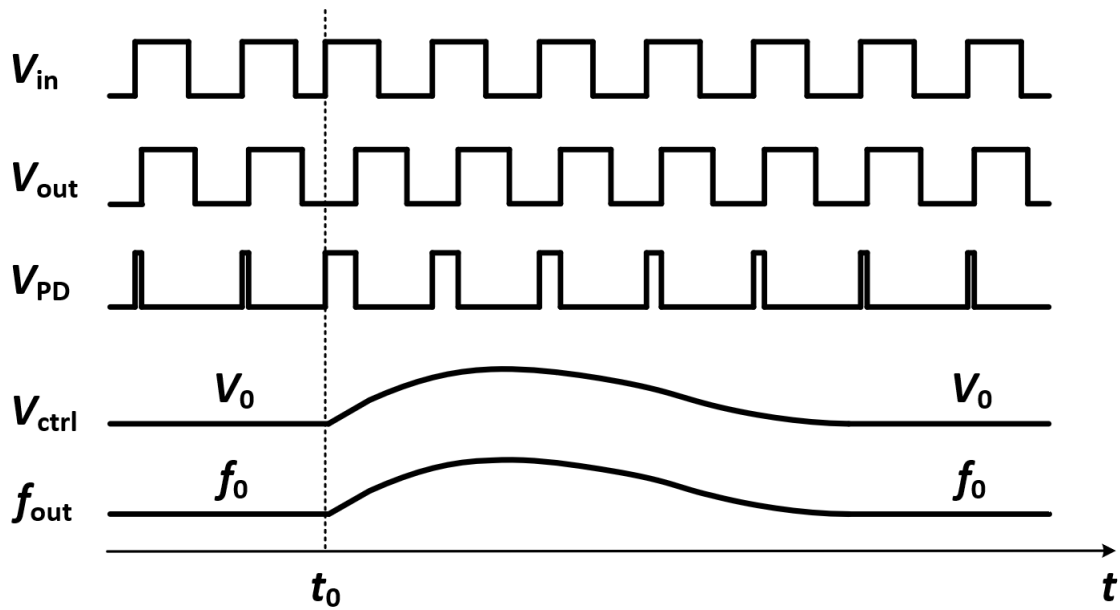


Figure 2.13: Apply a phase step at the input of a simple PLL

Note that whether applying a step change in frequency or phase, neither the control voltage nor output phase instantaneously jumps a step. Both frequency and phase changes occur gradually.

2.2 Transfer Function of Type-I PLL

Let's replace the basic PLL in Figure 2.11 with a linear representation shown as Figure 2.14. Here we use a simple first-order loop filter, and denote that $(1/RC) = \omega_{LPF}$.

The open-loop transfer function can be formulated as

$$L(s) = K_{PD} \times \frac{1}{1 + \frac{s}{\omega_{LPF}}} \times \frac{K_{VCO}}{s} \quad (2.18)$$

Equation (2.18) implies that there is only one pole at the origin of the open-loop transfer function. Therefore, the topology of Figure 2.11 is named after a "Type-I PLL".

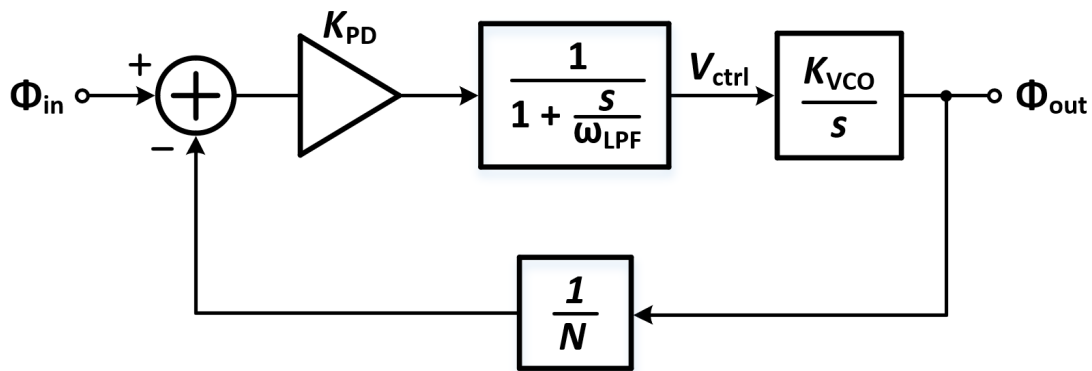


Figure 2.14: Linear model of a basic PLL

Next, we can derive the closed-loop transfer function as

$$H(s) = \frac{\phi_{out}}{\phi_{in}}(s) \quad (2.19)$$

$$= \frac{K_{PD}K_{VCO}}{\frac{s^2}{\omega_{LPF}} + s + \frac{K_{PD}K_{VCO}}{N}} \quad (2.20)$$

From the perspective of control theory, we usually express the denominator in the form $s^2 + 2\zeta\omega_n s + \omega_n^2$. Note that ζ is the damping factor and ω_n is the natural frequency, respectively. Rewrite the function below

$$H(s) = \frac{K_{PD}K_{VCO}\omega_{LPF}}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.21)$$

where

$$\zeta = \frac{1}{2} \sqrt{\frac{\omega_{LPF} N}{K_{PD} K_{VCO}}} \quad (2.22)$$

$$\omega_n = \sqrt{\frac{K_{PD} K_{VCO} \omega_{LPF}}{N}} \quad (2.23)$$



According to equation (2.22), we can deduce that the loop becomes more stable when N increases. A larger modulus N weakens the loop gain and increases the phase margin. On the other hand, a higher $K_{VCO} K_{PD}$ deteriorates the phase margin. The bode plot is illustrated in Figure 2.15. We also observe that the total phase will never be lower than -180° because there are only two poles in the open-loop transfer function: one at the origin and one at ω_{LPF} . This indicates that the type-I PLL system is unconditionally stable.

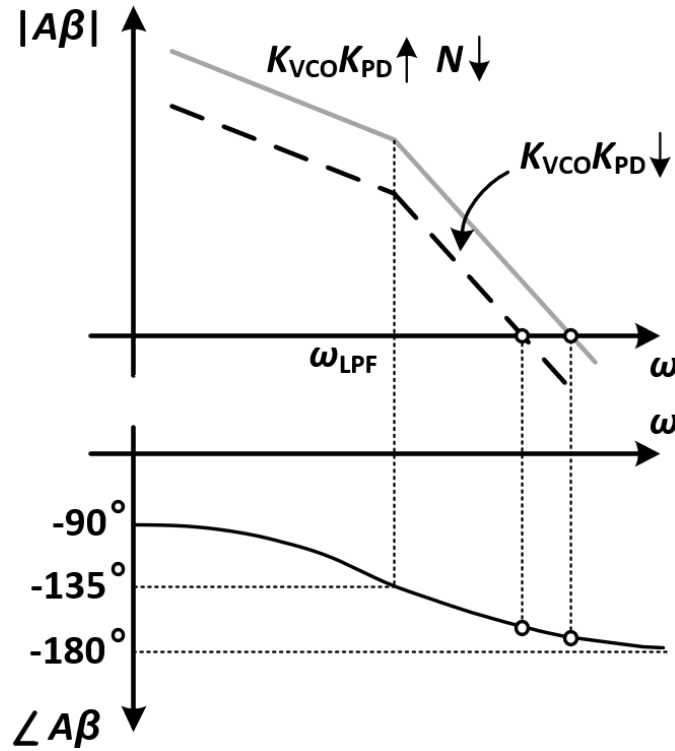


Figure 2.15: Bode plot of a basic type-I PLL

For the closed-loop analysis of type-I PLL, we start with the transfer function (2.21).

There is no zero in the system but two closed-loop poles located at

$$\omega_{p1,2} = (-\zeta \pm \sqrt{\zeta^2 - 1})\omega_n \quad (2.24)$$

For $\zeta = 1$, we have $\omega_{p1} = \omega_{p2} = -\omega_n$. When $\zeta^2 \gg 1$, then $\sqrt{\zeta^2 - 1} \approx \zeta[1 - 1/(2\zeta^2)] = \zeta - 1/(2\zeta)$, yielding

$$\omega_{p1} \approx -\frac{\omega_n}{2\zeta} = -\frac{K_{PD}K_{VCO}}{N} \quad (2.25)$$

$$\omega_{p2} \approx -2\zeta\omega_n = -\omega_{LPF} \quad (2.26)$$

Another aspect we are concerned with is the -3 -dB frequency, which indicates how much the input noise is filtered by the system. Let $|H(s = j\omega)| = 1/\sqrt{2}$, we obtain

$$\omega_{-3dB}^2 = [1 - 2\zeta^2 + \sqrt{(1 - 2\zeta^2)^2 + 2N^2 - 1}]\omega_n^2 \quad (2.27)$$

For simplicity, we set $M = 1$ here. It follows that

$$\omega_{-3dB} \approx 0.64\omega_n \quad \text{for } \zeta = 1 \quad (2.28)$$

$$\approx 0.37\omega_n \quad \text{for } \zeta = 1.5 \quad (2.29)$$

$$\approx 0 \quad \text{for } 2\zeta^2 \gg 1 \quad (2.30)$$

According to the calculation, for a type-I PLL, -3 -dB bandwidth decreases as the damping factor increases.

2.3 Limitation of Acquisition Range in Type-I PLL

In the previous analysis, we overlooked one important issue: Is the PLL guaranteed to lock once it is turned on? The answer is negative. If the VCO operates significantly far from the input frequency initially, the loop may fail to properly tune the VCO and consequently fail to lock to the input frequency in the end. The maximum initial value of $|f_{out} - f_{in}|$ for which the loop locks is called the "acquisition range".

To explain this phenomenon, let's recall 2.1.2. We mentioned that the XOR-based PD has K_{PD} of opposite polarities. This means that the PLL is no longer behaving as a linear system. If there is a frequency difference between the input signal and VCO output, the PD can transition between regions of different gain. This causes cycle slipping as shown in Figure 2.16. If the frequency difference is too large, the cycle slipping phenomenon may become more severe, and it may even lead to the PLL being unable to lock.

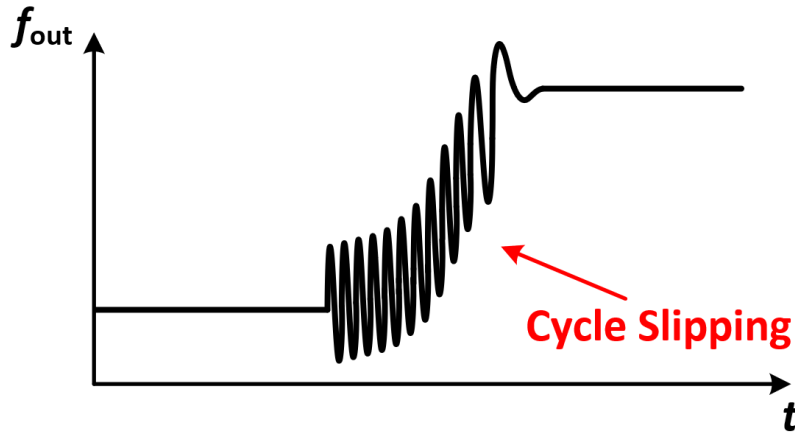


Figure 2.16: Cycle slipping

Looking at the acquisition range issue from a more rigorous perspective, let's first open the loop as illustrated in Figure 2.17. In the beginning, the VCO oscillates at $\omega_{out} \neq \omega_{in}$. The PD sense $V_0 \cos \omega_{in} t$ and $V_1 \cos \omega_{out} t$, producing $K_{PD} V_0 \cos (\omega_{out} \pm \omega_{in}) t$ at its output. After passing through the LPF, the high-frequency summing term is filtered out, leaving

only the $\omega_{out} - \omega_{in}$ term on V_{ctrl} .

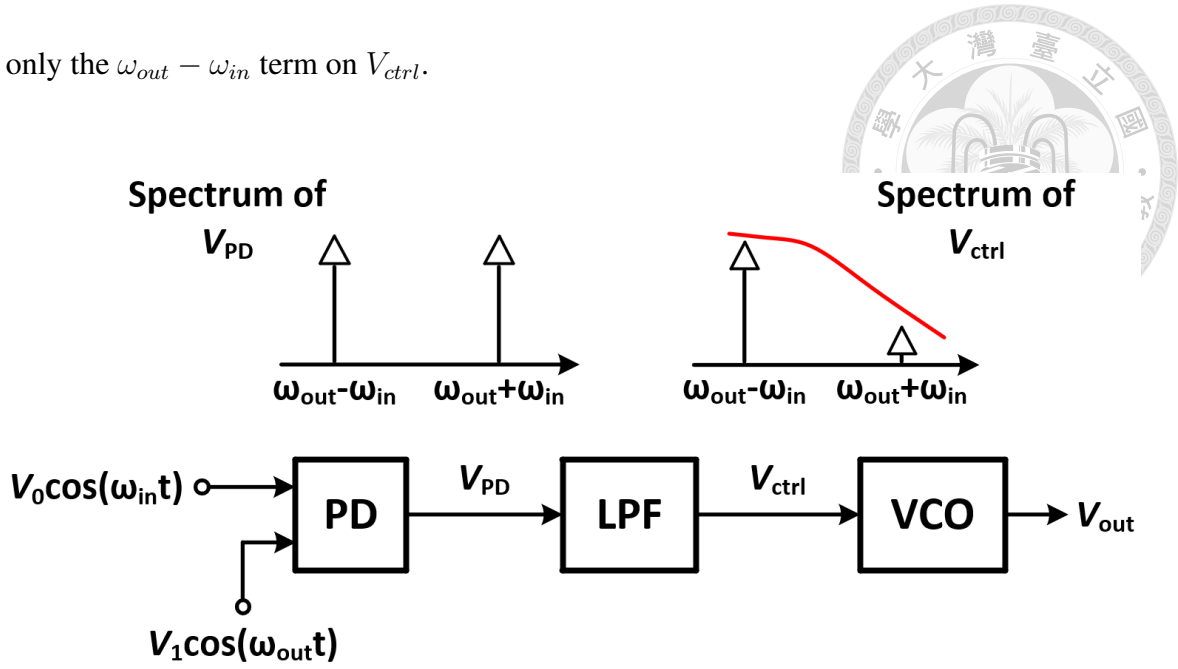


Figure 2.17: Open-loop PLL and spectrum of V_{PD} and V_{ctrl}

According to the narrowband FM approximation, the VCO output exhibits two sidebands at frequencies $\omega_{out} \pm \omega_m$ when there is a modulation signal $V_m \cos \omega_m t$ applied to its control line. In this scenario, where $\omega_m = \omega_{out} - \omega_{in}$, there are two sidebands present in the VCO output spectrum: ω_{in} and $2\omega_{out} - \omega_{in}$. As shown in Figure 2.18 (a), the DC value of V_{ctrl} is 0 now. Next, close the loop. The VCO output feeds back to the PD and mixes with the input signal again. The component at ω_{in} mixes with the input signal, also at ω_{in} , generating a DC value. This DC value drives the VCO frequency toward ω_{in} , and the PLL shows a tendency to lock. The progress of change is shown in Figure 2.18 (b).

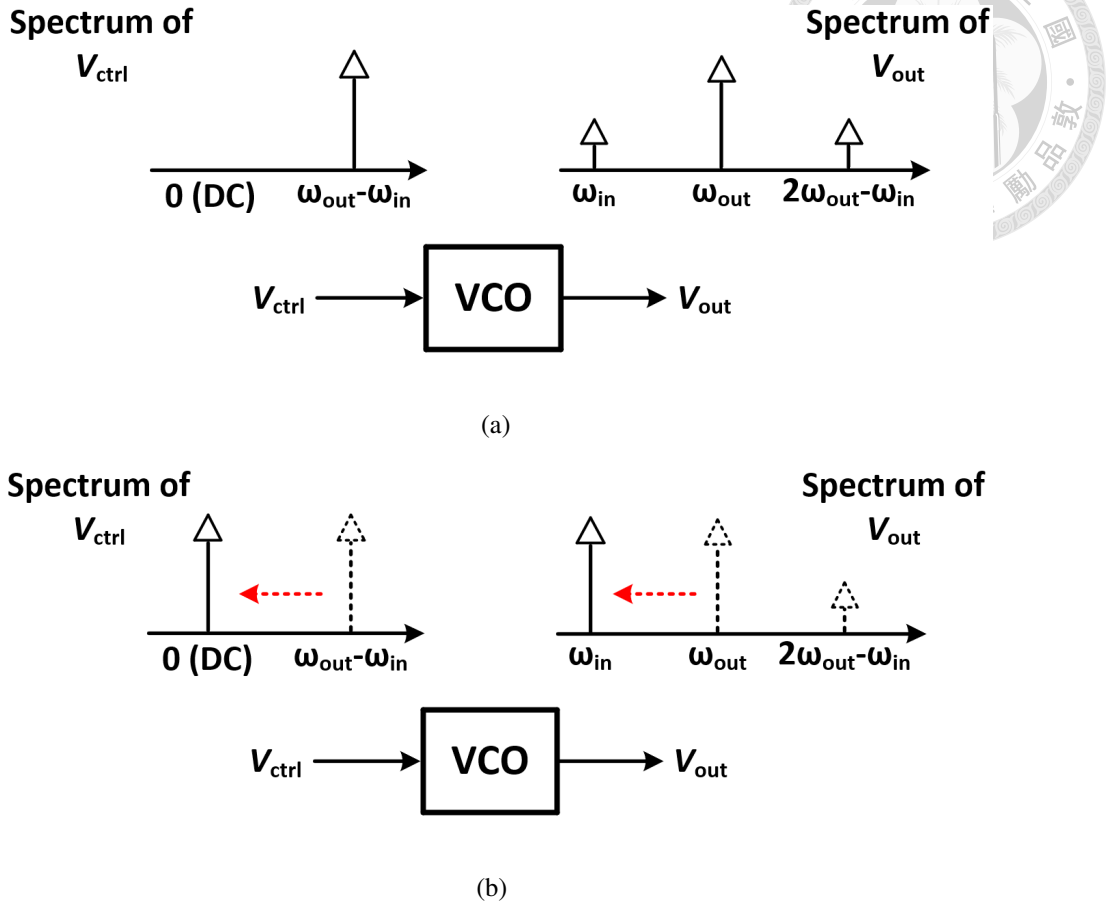


Figure 2.18: The locking progress

However, if the initial difference $|\omega_{out} - \omega_{in}|$ is too large, the tone at $\omega_{out} - \omega_{in}$ could be filtered by the LPF. Without a strong frequency component on V_{ctrl} , the modulation of the VCO would fail. This results in the VCO output tone at ω_{in} becoming too small. As a result, the DC component on V_{ctrl} becomes too narrow after mixing with the feedback signal to initial lock acquisition. We say that the PLL exceeds its acquisition range, as shown in Figure 2.19.

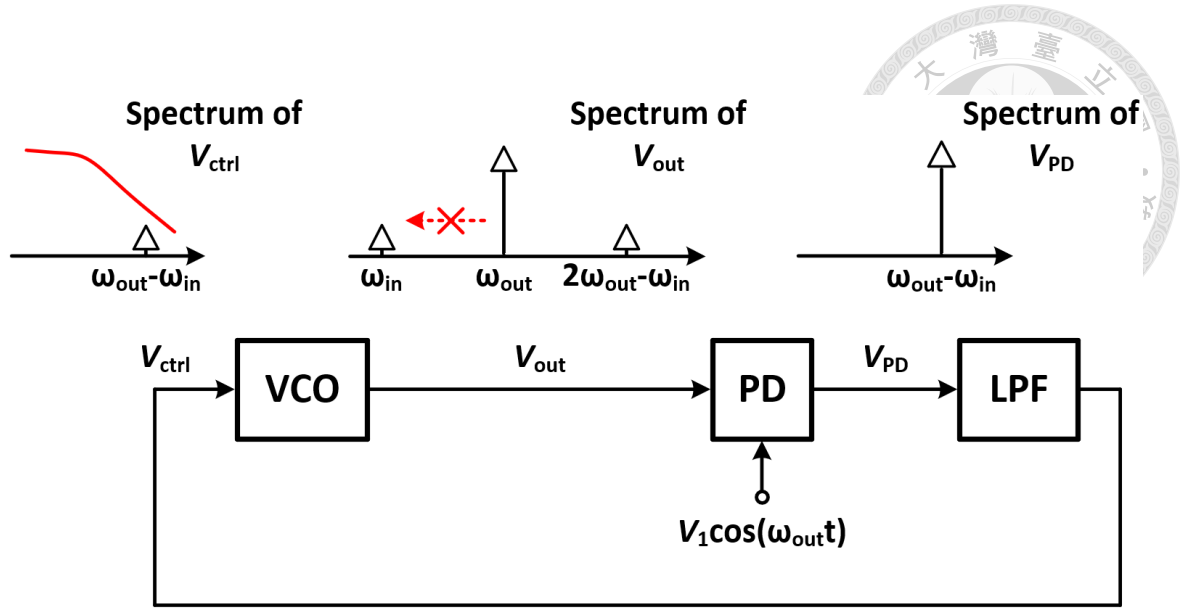


Figure 2.19: The PLL fails to lock if initial $|\omega_{out} - \omega_{in}|$ is too large

To alleviate this issue, increasing the bandwidth of the LPF is one approach. This prevents attenuation of the tone at $\omega_{out} - \omega_{in}$ and ensures normal modulation of the VCO. Besides, [15] offer an equation (2.31) to describe the input-referred acquisition range of a generic type-I PLL

$$\Delta\omega_L = \frac{2\pi K_{VCO} K_{PD}}{N} \quad (2.31)$$

where the input-referred frequency acquisition range is denoted as $\Delta\omega_L$.

Nevertheless, enlarging either the bandwidth of the LPF or the forward gain of the loop will increase the reference spur for a type-I PLL. All of these methods will widen the loop bandwidth, leading to a degradation in the reference spur. The relation between loop bandwidth and reference spur is discussed in [1], as shown in Figure 2.20.

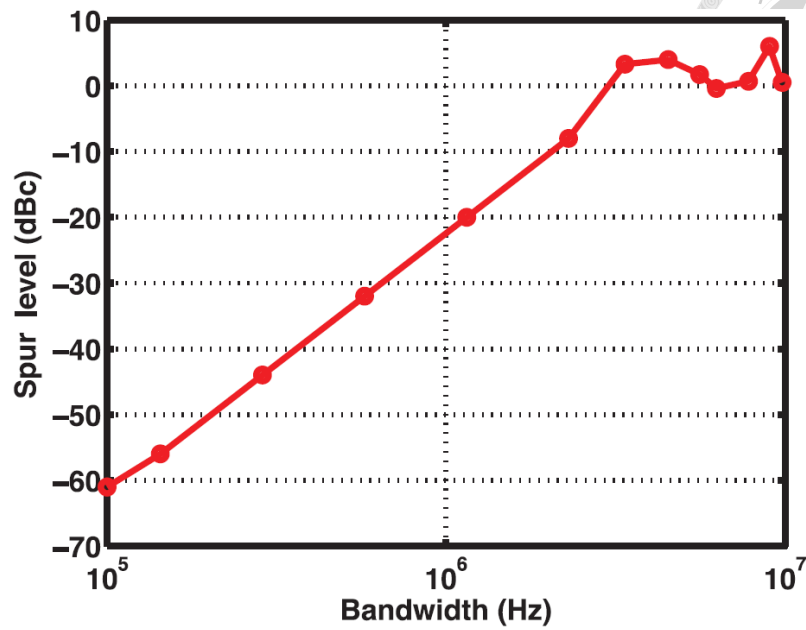


Figure 2.20: Reference spur versus bandwidth for type-I PLL [1]

In conclusion, the limited acquisition range severely restricts the practical application of type-I PLLs. Moreover, type-I PLLs suffer from a trade-off between the acquisition range and reference spur.



Chapter 3

Proposed PLL

3.1 Overview

This chapter introduces a wide output frequency range type-I PLL. The overall architecture will be deliberated first, and a band-selecting mechanism will be proposed to solve the problem of acquisition range in type-I PLL architecture. Subsequently, the chapter delves into the detailed circuit implementations. Finally, the overall system simulation results are demonstrated to validate our design.

3.2 Overall Architecture

As mentioned in 2.3, type-I PLL encounters limitations in its acquisition range. This issue significantly constrains the practical viability of the type-I PLL architecture. Especially in applications involving wide output range frequency synthesizers, the limited acquisition range may cause the loop to be unlocked easily.

Here is an example of Figure 3.1 that illustrates this problem. Let's set the reference frequency $f_{REF} = 10$ MHz and divisor $N = 200$, thus the output frequency f_{out} will be

locked at 2 GHz. If a user wants to switch the output frequency to 1 GHz, the divisor should be changed to $N = 100$ without altering the reference frequency. However, due to the gradual frequency changes in a type-I PLL, f_{out} does not drop to 1 GHz as soon as switching divisor. Instead, it stays around 2 GHz and approaches 1 GHz gradually. This results in the feedback frequency $f_B = f_{out}/N \approx 20$ MHz at the moment of divisor switching. At this point, PD receives a significant difference between the two input frequencies f_{REF} and f_B , which exceeds the acquisition range and leads to PLL unlocking. Figure 3.1 presents the transient simulation results of the scenario as mentioned earlier, clearly illustrating the phenomenon of the PLL unlocking.

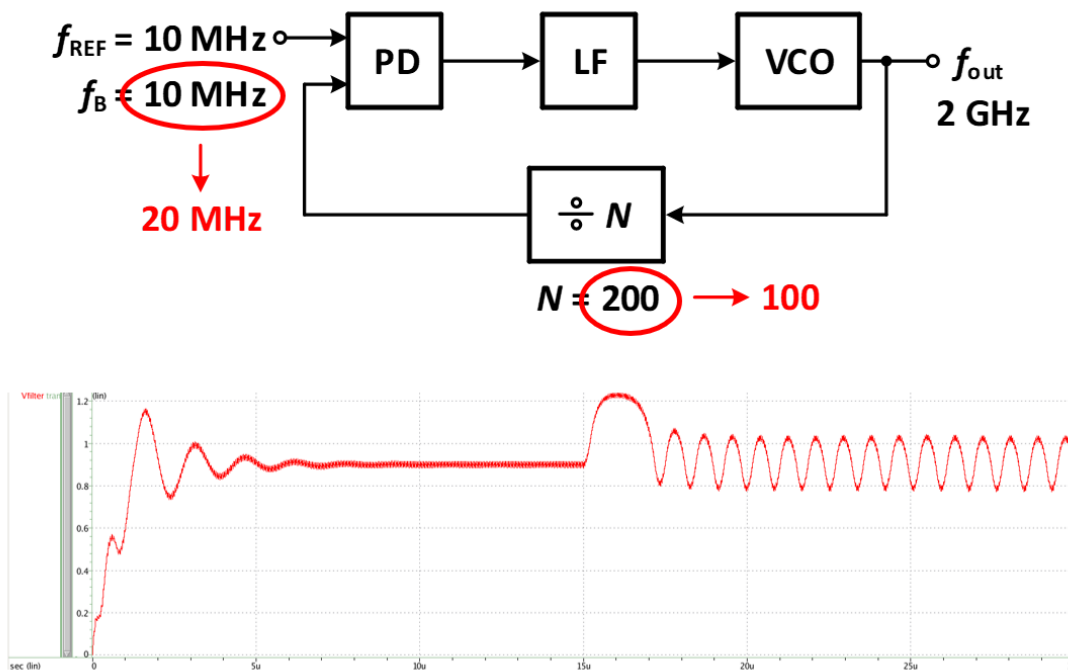


Figure 3.1: Unlocking phenomenon when switching divisors

If we want to expand the output frequency range, the acquisition range must be widened. A common practice to expand the PLL's acquisition range is to increase the loop bandwidth. However, with the increase in bandwidth, reference spur also increases correspondingly,

and there are limits to how this method can enhance the acquisition range.

To solve this problem, we propose a new approach to achieve a wide-band output. As the divisor changes, we can make the output frequency of VCO jump to a frequency close to the target frequency first. Within the acquisition range, we can prevent the unlocking phenomenon. Subsequently, the characteristics of PLL are utilized to lock it to the target frequency gradually and converge at a stable state eventually. Taking Figure 3.2 as an example, when switching from $N=200$ to $N=100$, the output frequency of the VCO is switched to a frequency close to 1 GHz simultaneously. This ensures that the frequency fed back to the phase detector does not deviate too much from the reference frequency, and avoids the limitations of the type-I PLL acquisition range. Note that we haven't altered the loop bandwidth through this method, so the spurs would not degrade. This breaks the trade-off between output frequency range and reference spur level.

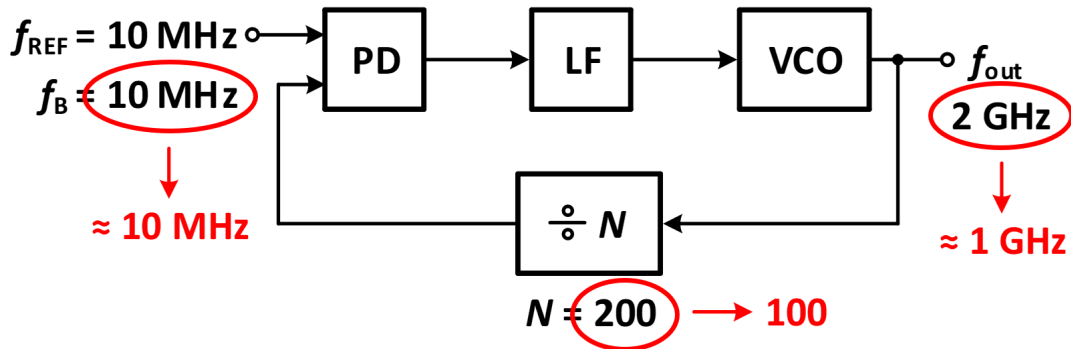


Figure 3.2: Proposed approach to avoid the limitations of acquisition range

In order to perform an instant frequency switching, we break the VCO tuning range down to several frequency bands. Now, we need to consider the methods for controlling the bands of this multi-band VCO. One realization is to map the frequency band to the corresponding divisor, enabling the control signal of the divider to determine the desired band at the same time. However, this approach couldn't cope with the impact of PVT

variations. Once the output frequency deviates due to the variations, the initially set frequency band may no longer include the target frequency. Another commonly used implementation, depicted in Figure 3.3, utilizes two comparators, a logic circuit, and a counter to generate control bits for VCO bands. During the process of loop locking, V_{ctrl} rises along A_1 , causing f_{out} to increase accordingly. If the target frequency f_1 is not reached within the range of $V_{min} < V_{ctrl} < V_{max}$, the two comparators generate $D_1D_2 = 00$ or $D_1D_2 = 11$, and the counter increments by 1, causing the VCO to switch to the next frequency band. Subsequently, the loop attempts to lock again, continuously searching until it switches to a feasible frequency band, ensuring that the loop is locked within the range of $V_{min} < V_{ctrl} < V_{max}$.

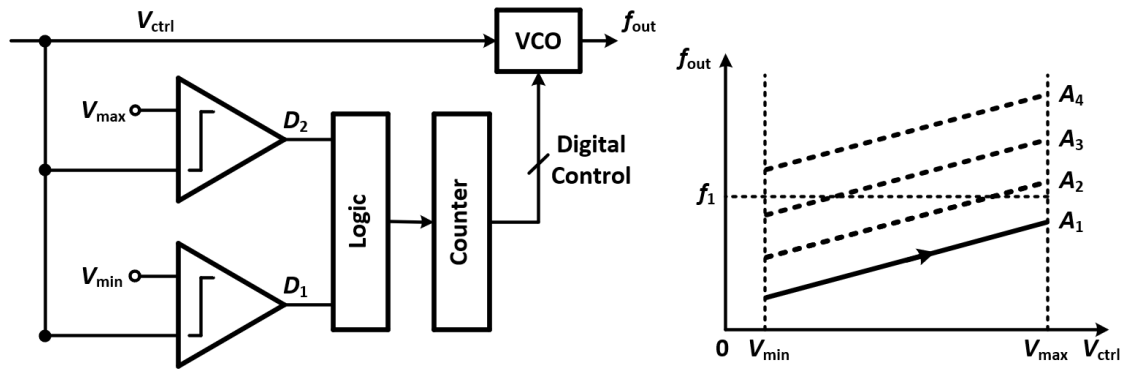


Figure 3.3: The conventional method of frequency band controlling

However, this technique couldn't be used in type-I PLL architecture. The critical factor is that the system is constrained by the limited acquisition range, leading to the inability of the loop to lock in the case of progressive frequency modulation.

To solve this problem, we propose a novel band-selecting mechanism. While the divisor is changed, VCO will be disconnected from V_{ctrl} and the system enters a "band-selecting" process, which is illustrated in Figure 3.4. The output frequency of the VCO will be monitored and compared with the target frequency. If the difference between the two

exceeds an acquisition range, then updates a new set of control bits and switches to a band closer to the target frequency. This process continues until the two frequencies are within the acceptable range, at this point, the band-selecting process ends, and the PLL reconnects. Through the strategy we propose, VCO is allowed to rapidly jump to a value near the target frequency before the main loop initiates the locking process. This breakthrough effectively overcomes the limitation of the acquisition range in Type-I PLL.

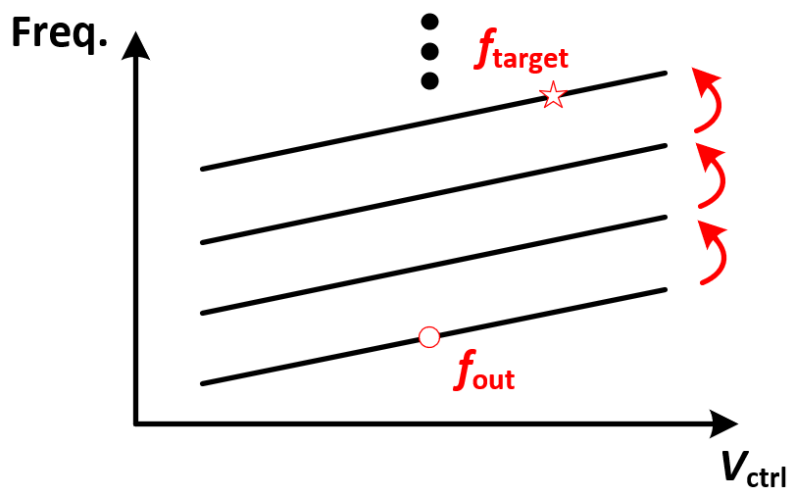


Figure 3.4: The proposed band-selecting process

Figure 3.5 is the block diagram of the frequency synthesizer proposed in this paper. This PLL is designed to generate a total of 36 different frequencies ranging from 840 MHz to 2240 MHz, with a planned reference frequency of 10 MHz. In addition to the main loop composed of PD, LF, multi-band VCO, and multi-modulus divider, we will utilize an auxiliary loop called "Band-Selecting Loop (BSL)" to execute the band-selecting process and generate control bits for the multi-band VCO. In the next section, I will provide a detailed explanation of the proposed band-selecting mechanism and its implementation. The circuit architectures of other blocks will also be presented subsequently.

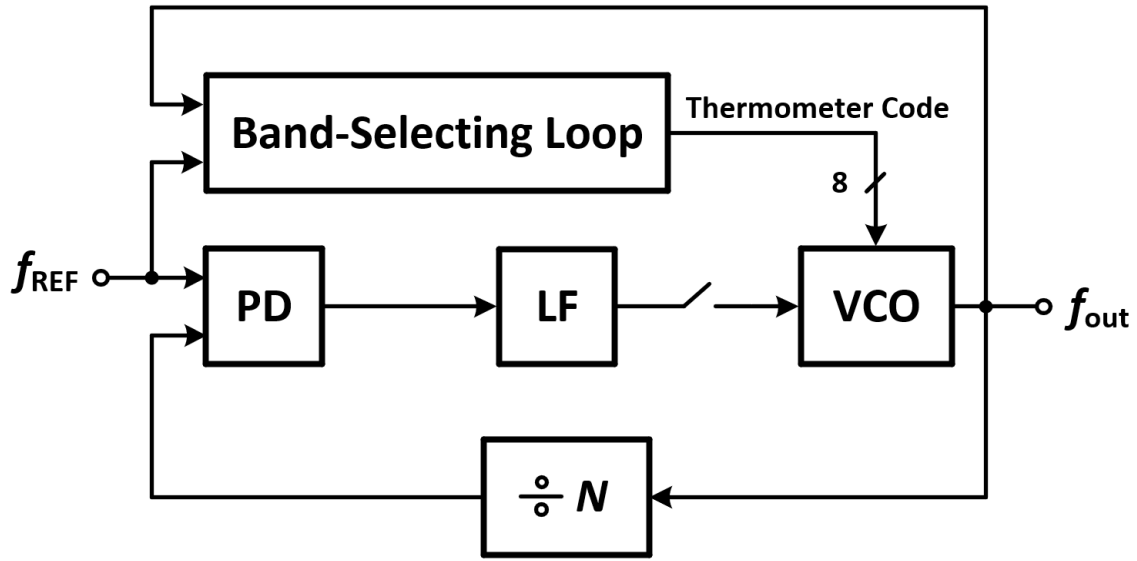


Figure 3.5: The block diagram of the proposed wide frequency range Type-I PLL

3.3 Proposed Band-Selecting Loop

3.3.1 Band-Selecting Mechanism

Figure 3.6 is the finite-state machine of the proposed band-selecting mechanism. Before we get started, let's revisit the example we mentioned in Figure 3.1. We aim to address the unlocking issue caused by a significant frequency difference between the two inputs of PD when the divisor is switched. Thus, in state 0, we need to sense the control bits of the divider. As soon as the control signal switches, that is, when the PLL divisor changes, disconnect the main loop and activate BSL to start the band-selecting process. The first step is to obtain the current output frequency of VCO, denoted as f_{out} in the following description. Next, transition to state 2, compare the obtained f_{out} with the target frequency, denoted as f_{tar} . If the difference between the two exceeds an acquisition range, transition to state 3, update the control signal of VCO and switch to the next frequency band. If the difference is less than an acquisition range, it indicates that the current frequency band is

correct. Therefore, BSL will be shut down and reconnect the main loop, completing the band-selecting process. In the following paragraphs, we will discuss how to implement each state by actual circuits.

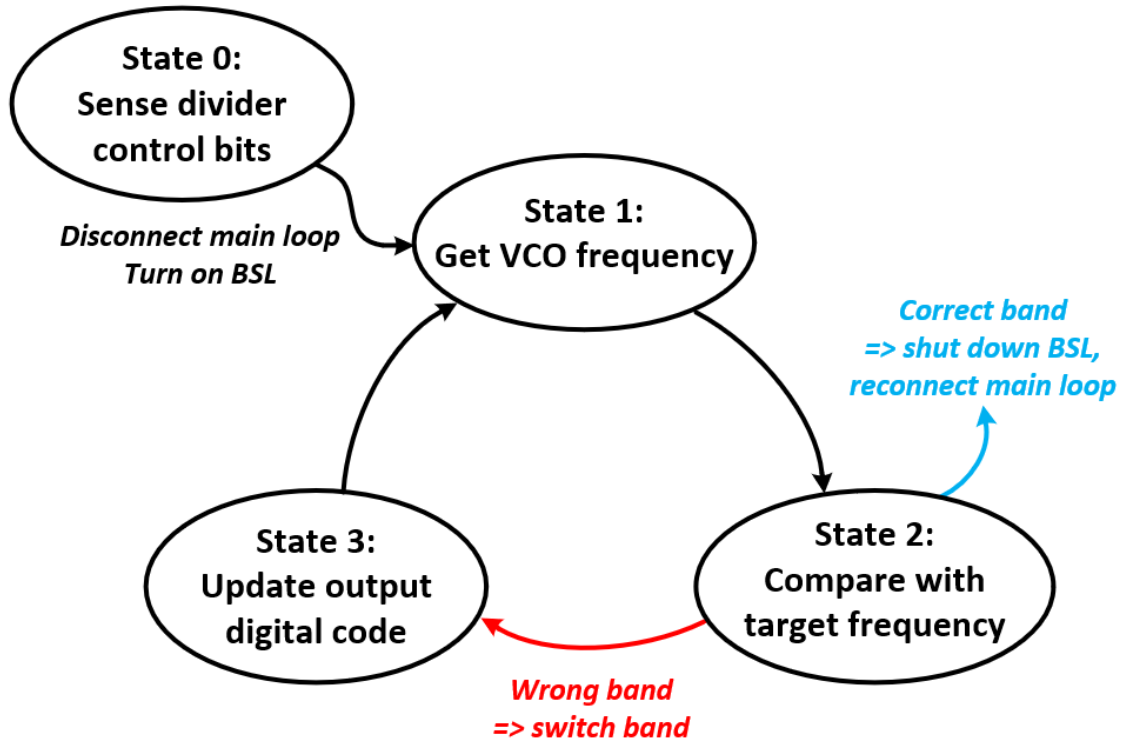
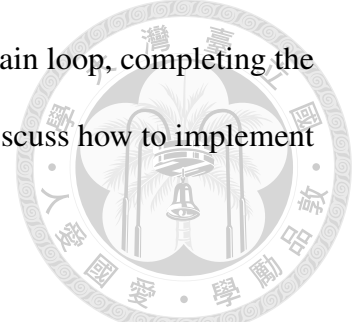


Figure 3.6: The FSM of the proposed band-selecting mechanism

- State 1:

The purpose of this state is to sense VCO frequency. To get the value of f_{out} , we can think of the relation between VCO output and reference signal. As shown in Figure 3.7, if the period of V_{out} is T_{out} , and the period of V_{REF} is T_{REF} , there will be T_{REF}/T_{out} VCO cycles in one reference period. After a simple calculation,

$$\frac{T_{REF}}{T_{out}} = \frac{f_{out}}{f_{REF}} \quad (3.1)$$

we can get the ratio between f_{out} and f_{REF} . For example, assume $f_{REF} = 10$ MHz

and $f_{out} = 1000 \text{ MHz}$, then we can count to $1000/10 = 100$ VCO cycles in one reference period.

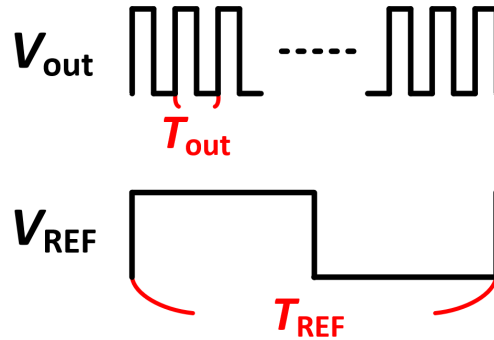


Figure 3.7: The relation between VCO output and reference signal

As mentioned above, by counting the amount of VCO cycles in one reference period, the information of f_{out} can be obtained. Thus, the idea is to use a counter to execute this task. Figure 3.8 shows the concept. Let VCO output signal V_{out} be the input of the counter and reference be the reset signal. The counter will reset every T_{REF} . We can access the desired value f_{out}/f_{REF} through this method.

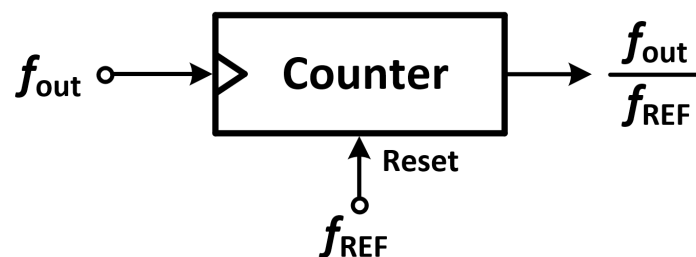


Figure 3.8: Use a counter to get the ratio of f_{out} and f_{REF}

- State 2:

In this state, we will compare f_{out} with the target frequency f_{tar} . We have obtained f_{out}/f_{REF} in state 1, so let's see how this parameter is related to f_{tar} . Here we define two values first, where R represents the value of the acquisition range, and N

represents the divisor of the PLL. Note that N happens to be the result of dividing the target frequency by the reference frequency. To avoid unlocking, we need to check if the difference between f_{out} and f_{tar} is within an acquisition range. Which can be formulated as

$$|f_{out} - f_{tar}| < R \quad (3.2)$$

Thus, the acceptable range of f_{out} can be arranged as

$$f_{tar} - R < f_{out} < f_{tar} + R \quad (3.3)$$

Dividing by f_{REF} , we can obtain the following equation:

$$\frac{f_{tar}}{f_{REF}} - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < \frac{f_{tar}}{f_{REF}} + \frac{R}{f_{REF}} \quad (3.4)$$

While $f_{tar}/f_{REF} = N$, finally we have

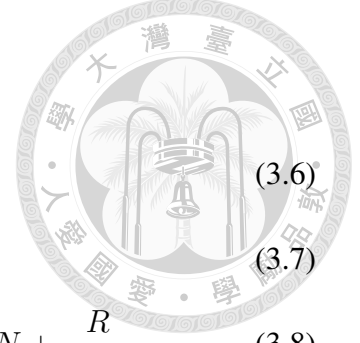
$$N - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < N + \frac{R}{f_{REF}} \quad (3.5)$$

Observing (3.5), it can be noticed that f_{out}/f_{REF} here is the output obtained by the counter mentioned in state 1. These equations imply that if f_{out}/f_{REF} is between $N - R/f_{REF}$ and $N + R/f_{REF}$, then we can confirm that the comparison result between f_{out} and f_{tar} falls within the acquisition range.

The implementation involves using two comparators to compare the relation between f_{out}/f_{REF} and $N \pm R/f_{REF}$. The block diagram is shown in Figure 3.9. There are

three possible results:

$$X_1X_0 = \begin{cases} 11, & \frac{f_{out}}{f_{REF}} > N + \frac{R}{f_{REF}} \\ 00, & \frac{f_{out}}{f_{REF}} < N - \frac{R}{f_{REF}} \\ 01, & N - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < N + \frac{R}{f_{REF}} \end{cases} \quad (3.6)$$



According to the equations, if f_{out} is higher than acquisition range, then output $X_1X_0 = 11$. If f_{out} is lower than acquisition range, then output $X_1X_0 = 00$. If f_{out} is within the acquisition range, then output $X_1X_0 = 01$. In other words, the output X_1X_0 indicates whether the current band is too high, too low, or correct.

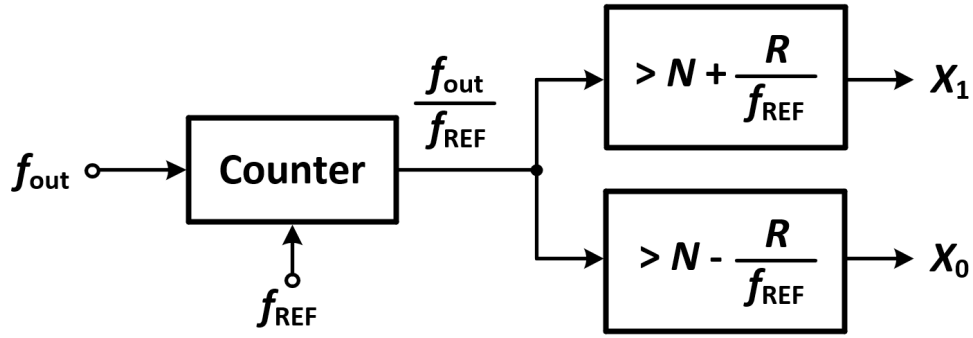


Figure 3.9: Use two comparators to implement state 2

- State 3:

When the output of the previous state shows that f_{out} is out of the acquisition range, which is $X_1X_0 = 11$ or $X_1X_0 = 00$, the control bits of VCO band need to be updated to make f_{out} get closer to f_{tar} . In this state, a bi-directional shift register is applied to produce a set of thermometer codes that control the VCO band. As depicted in Figure 3.10, if $X_1X_0 = 11$, the bi-directional shift register will produce an additional low output, causing the VCO to shift down one band. If $X_1X_0 = 00$, the bi-directional shift register will produce an additional high output, causing the

VCO to shift up one band.

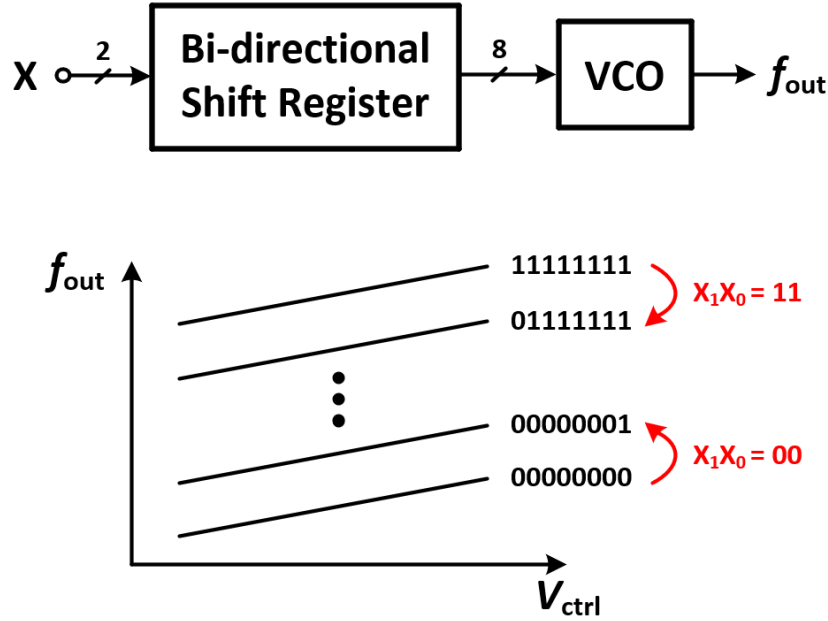


Figure 3.10: Update control bits by a bi-directional shift register

3.3.2 Clock Generator

We have elaborated on the concept and the implementation of the band selection mechanism. In the discussion above, we can observe that the operations of the three states follow a specific order. Therefore, the challenge we face next is how to correctly clock each state, enabling them to operate sequentially and efficiently. Note that considering the goal of cost savings in practical applications, we aim to avoid the use of an external signal generator, such as a quartz oscillator, to generate clock signals for BSL. Instead, we will generate clock signals within the chip. The only external signal is the reference signal shared with the entire PLL.

Figure 3.11 illustrates the timing diagram of the proposed band-selecting mechanism. f_{REF} is regarded as the reference. Assuming we start state 1 from t_1 , calculate the number of cycles for f_{out} . After one reference period, which is at time $t_1 + T_{REF}$, we can obtain

the result of f_{out}/f_{REF} and compare it with the upper bound and lower bound in state 2. At the right next falling edge, update the control bits for VCO bands in state 3 based on the result of the previous comparison.

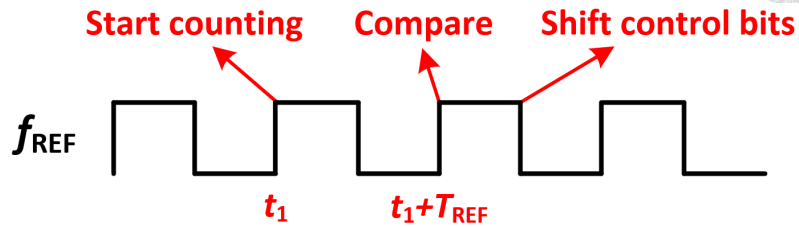


Figure 3.11: Sequence of each state in proposed band-selecting mechanism

According to what was mentioned earlier, Figure 3.12 shows the three clock signals we need. The rising edge of each clock will trigger the state respectively. In this way, we can ensure that BSL operates properly. We will design a clock generator that produces these signals.

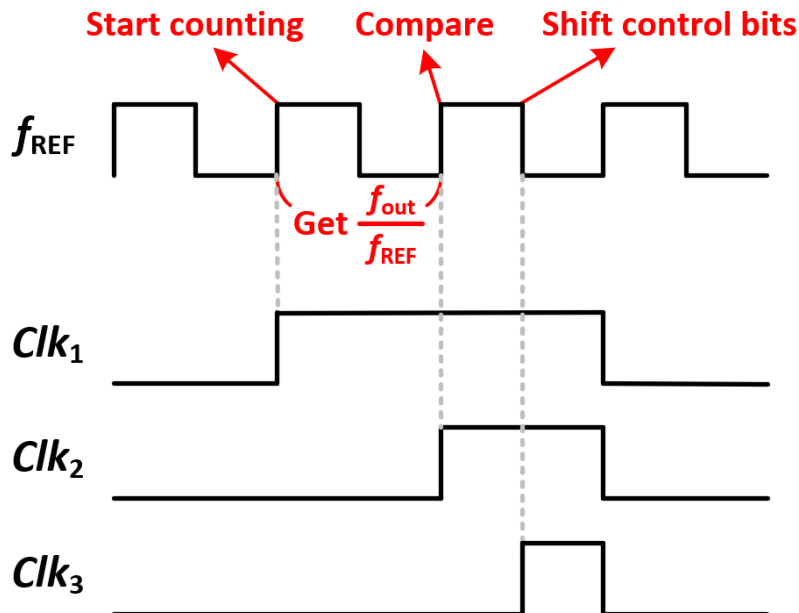


Figure 3.12: Three clock signals for respective state

The required clock signals can be generated by a synchronous counter. The structure is

depicted in Figure 3.13. Note that the last D flip-flop which produces Clk_3 is a falling-edge triggered DFF, and there is an additional DFF in the first stage to ensure the reliability of this clock generator. The schematic of DFF is shown in Figure 3.14

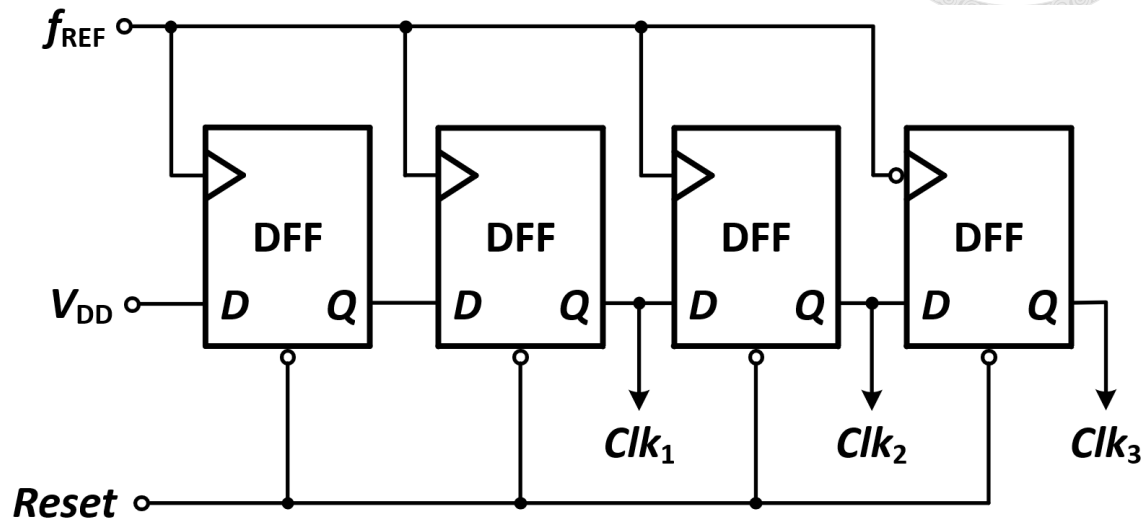


Figure 3.13: Realize the clock generator by a synchronous counter

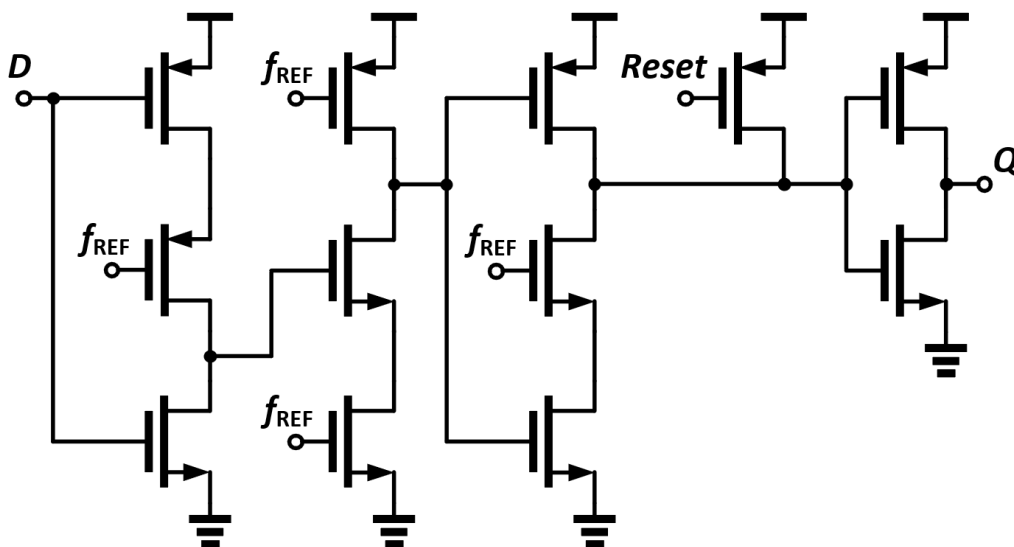


Figure 3.14: schematic of D flip-flop

Let's take a closer look at the clock generator. The outputs of the synchronous counter $Clk_{1,2,3}$ will stay *HIGH* once they switch their state. Hence we need to reset it every cycle of band-selecting. Clk_3 is responsible for clocking the last state in the band-select

mechanism, so we use Clk'_3 (the reset of DFF is negative-triggered) to generate the reset signal. Besides, a DFF is placed after Clk'_3 to generate the reset signal with a half-cycle delay, ensuring that state 3 has been completed before resetting the clock generator. Figure 3.15 illustrates the integral timing diagram. Note that a complete band-selecting cycle requires the duration of $4 * T_{REF}$, which is 0.4 usec in this work.

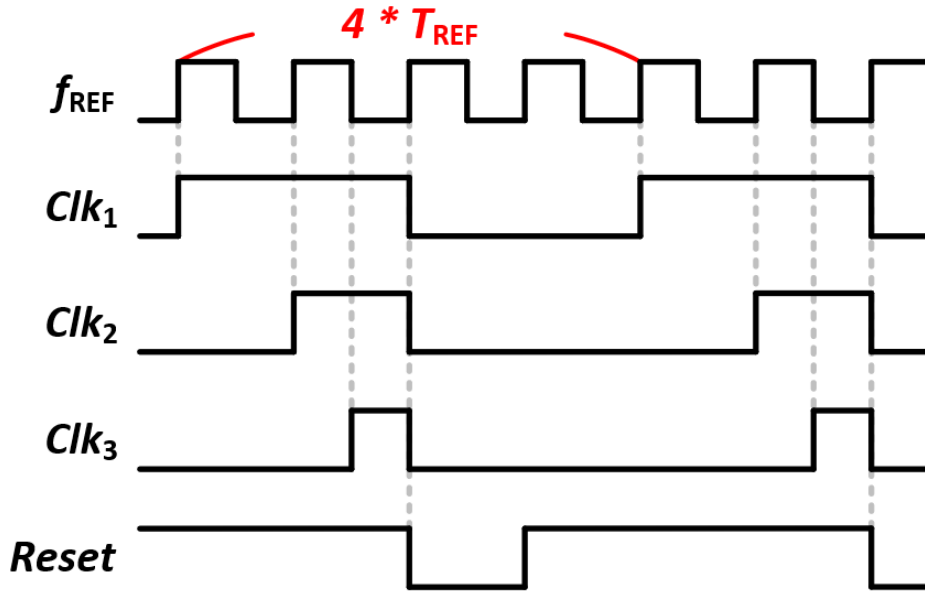


Figure 3.15: The complete timing diagram of clock generator

Earlier, we have discussed the situations when the band is too high or too low. Now, let's consider the case when f_{out} falls exactly within the acquisition range, i.e., when $X_1X_0 = 01$. With the goal of cost saving, we intend to halt the operation of BSL while reconnecting to the main loop.

As the comparators produce two opposing outputs when the band is correct, we can use an XNOR gate to detect X_1X_0 .

$$\text{XNOR output} = \begin{cases} 0, & X_1X_0 = 01 \\ 1, & X_1X_0 = 00, 11 \end{cases} \quad (3.9)$$

$$(3.10)$$

Utilizing the property of XNOR, we combine its output as the reset signal for the clock

generator. The implementation is shown in Figure 3.16, where an AND gate is added after the DFF mentioned earlier. When XNOR outputs 1, it does not affect the reset signal. However, when XNOR outputs 0, indicating that the current band is correct, it forcibly resets the clock generator, causing $Clk_{1,2,3}$ to cease generation. Every block in BSL subsequently stops functioning, thereby achieving the goal of reducing power consumption.

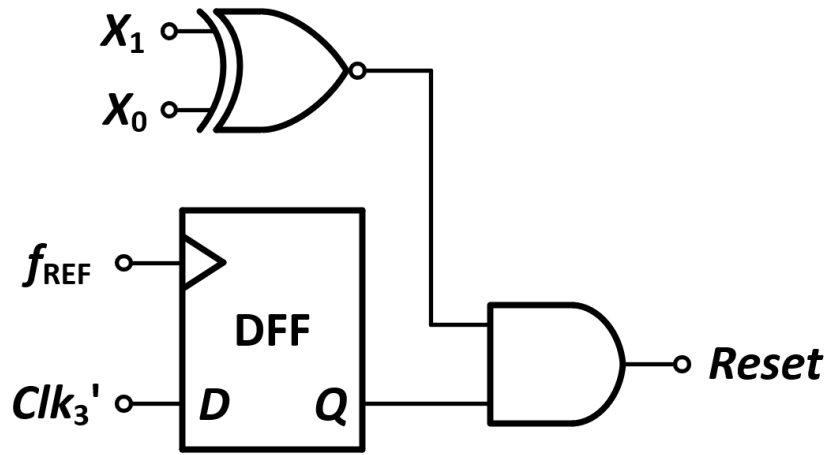


Figure 3.16: Reset signal in clock generator

Now we can calculate the time required for the band-selecting process. Recall Figure 3.12, we need $4 \times T_{REF}$ seconds to complete one band switch. For the cycle when the band is already in the acceptable range, the band-selecting mechanism starts from Clk_1 rises. As soon as the comparators generate $X_1X_0 = 01$ at Clk_2 rises, the clock generator is turned off and the band-selecting process ends. In summary, the last cycle only takes one T_{REF} , and the time required to switch A bands can be formulated as:

$$\Delta t = A \times T_{REF} + T_{REF} \quad (3.11)$$

3.3.3 Activation of Band-Selecting Loop

In 3.3.1, we mentioned that we need to monitor the modulus of the divider to determine whether to activate BSL. In this section, we will discuss the implementation of state 0 in detail.

Recall that by disabling the clock generator with the reset signal, we can stop the operation of BSL. And its reset signal is controlled by the XNOR result of X_1X_0 . Therefore, the idea is that when a divisor switch is detected, forcefully changing X_1X_0 from 01 to 00. So that the reset signal of the clock generator is no longer locked at 0, allowing it to restart the generation of $Clk_{1,2,3}$, thus restarting the operation of BSL.

Accordingly, we need an edge detector to monitor the control bits of divider D_{in} first. Once the control signal of the divider switches, it generates a pulse. Then use the edge detector's output as the reset signal for comparators. In this way, when a pulse is generated, it resets comparators' outputs to $X_1X_0 = 00$. As a result, the clock generator starts working and activates BSL. The block diagram is shown in Figure 3.17.

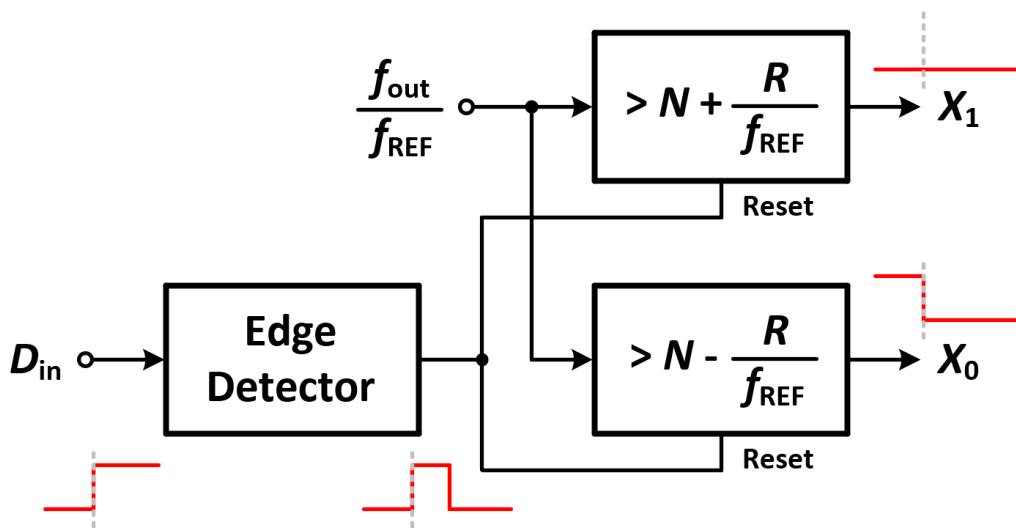


Figure 3.17: Implementation of state 0

Note that since the divisor doesn't change precisely at the rising edge of the reference,

when the edge detector generates a pulse, in the worst-case scenario, it will require a time of T_{ref} before the clock generator generates Clk_1 again. So we need to modify formula (3.11) to

$$A \times T_{REF} + T_{REF} < \Delta t < T_{REF} + A \times T_{REF} + T_{REF} \quad (3.12)$$

3.3.4 The Whole Structure of Band-Selecting Loop

In this section, we will integrate the above discussions to present a complete architecture of BSL. But before we start, there are still some details that need to be supplemented. Equation 3.5 gives us inspiration for implementing state 2. However, we have not yet delved into the process of determining and generating the values for the upper and lower limits, which serve as references for the comparators.

While N is decided by the control bits of the divider, R/f_{REF} is the value that needs to be defined. In the previous simulation shown in Figure 3.1, we observed instances of unlocking when the difference between f_{out} and f_{tar} exceeded 200 MHz. So, we can state that the acquisition range is 200 MHz within this bandwidth condition. To ensure the correctness of the circuit after tape-out, we have incorporated some margin by setting $R = 160$ MHz. Henceforth, we can conclude that $R/f_{REF} = 16$, which is a fixed value in the system.

To calculate $N \pm R/f_{REF}$ in the circuit, we adopt two digital adders. One performs $N + R/f_{REF}$ directly, the other adds N with the complements of R/f_{REF} to achieve $N - R/f_{REF}$. The outputs are two series of digital signals, which will serve as the comparison reference values for the comparators.

Now we can illustrate the full picture of the proposed band-selecting loop, as depicted in Figure 3.18. The loop initiates with an edge detector sending a pulse to reset comparators

upon sensing a change in the digital control signal of the divider. As X_1X_0 resets to 00, thus *Power* signal becomes *HIGH* and turn on the clock generator. Subsequently, with the aid of $Clk_{1,2,3}$, counter, comparators, and bi-directional shift register operate sequentially. When a band-selecting cycle is completed, a set of control bits is generated as the control signal for the VCO band. The band-selecting process continues until $X_1X_0 = 01$, signaling that f_{out} and f_{tar} have fallen within an acquisition range. At this point, *Power* becomes *LOW*, deactivating the clock generator and concluding the band-selecting process.

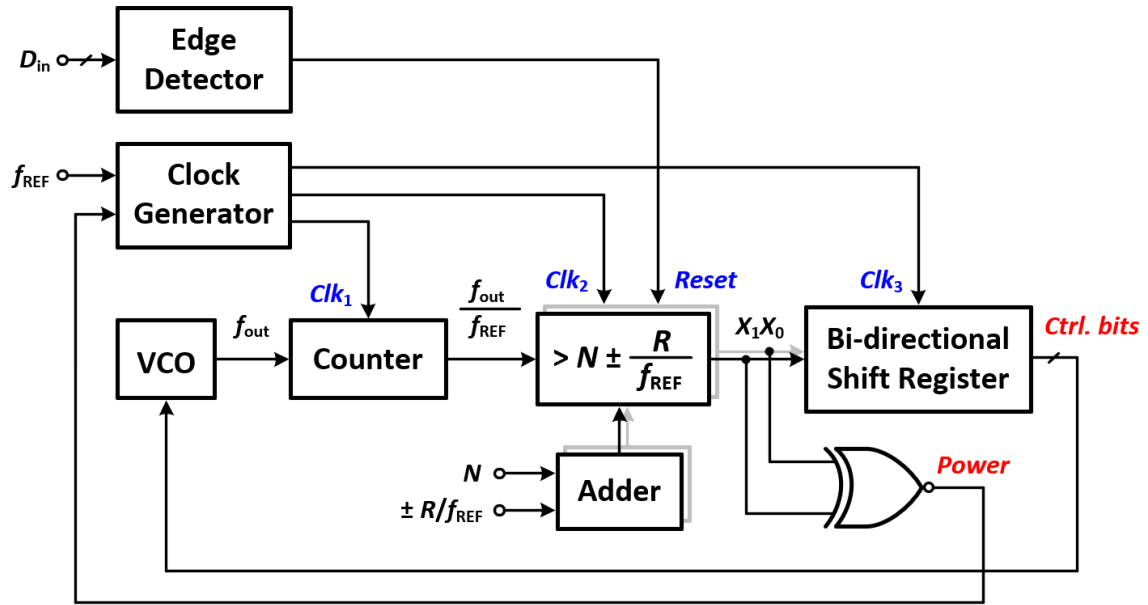


Figure 3.18: The full picture of the proposed BSL

The simulated transient response of the proposed BSL is shown in Figure 3.19. The divisor is adjusted at two points, $t = t_1$ and $t = t_2$. The edge detector produces a pulse to reset the output of comparators X_1X_0 , activating the power and initiating the band-selecting process.

At $t = t_1$, the divisor switches, leading to a transition in the target frequency from 1200 MHz to 2000 MHz. At $t = t'_1$, the BSL detects that $|f_{out} - f_{target}|$ is within the predefined range of 160 MHz, indicating that the desired frequency has been reached. Subsequently,

or switches and lets the
es the same process, but
range and then ends the

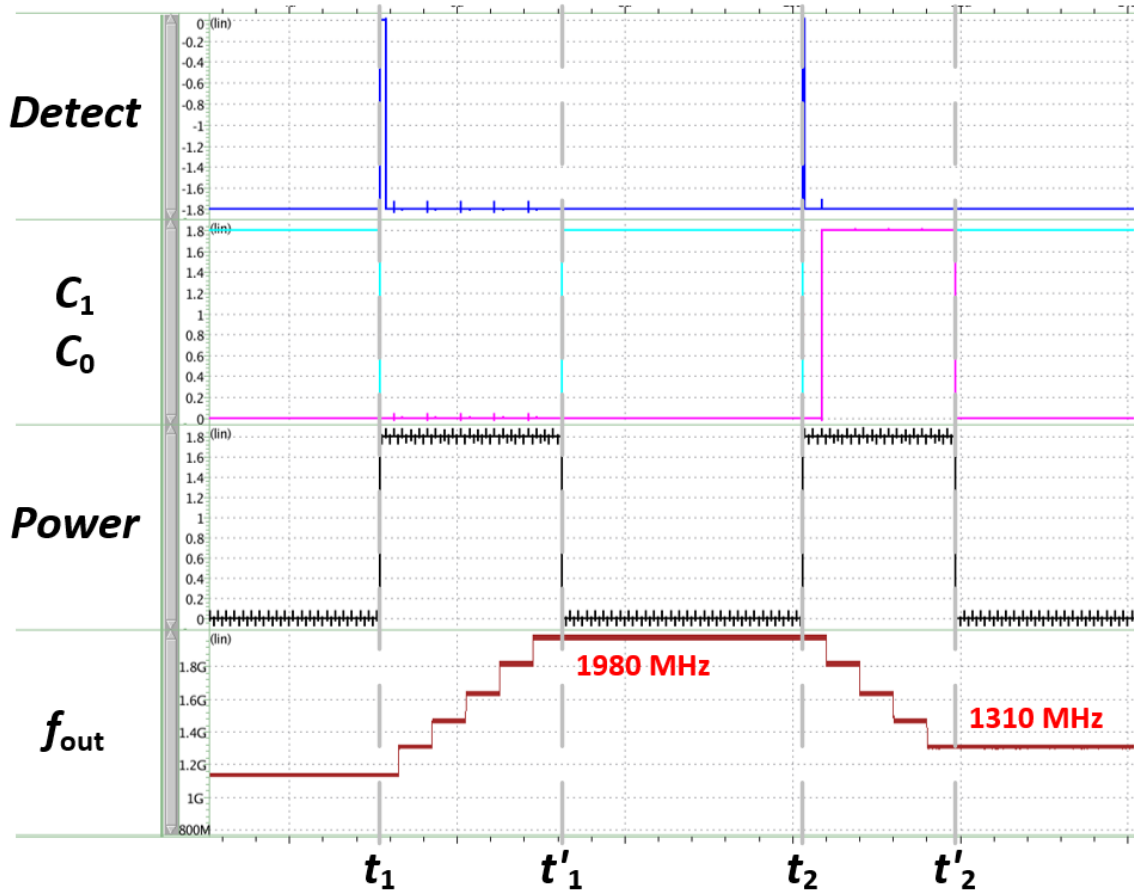


Figure 3.19: Transient simulation result of BSL

3.4 Circuit Implementation

In this section, we will present the implementation of each block in the proposed PLL. Begin with the circuit in the main loop and conclude by revisiting the band-selecting loop, discussing the relation between BSL and other blocks.

3.4.1 Phase Detector and Master-Slave Sampling Filter

The PD in this work is implemented using an XOR gate. To reduce the large ripple on V_{ctrl} caused by the XOR gate, we adopt a discrete-time filter instead of an analog RC filter. Referring to the approach introduced in [1], we employ a master-slave sampling filter (MSSF), as illustrated in Figure 3.20, to enhance the reference spur performance.

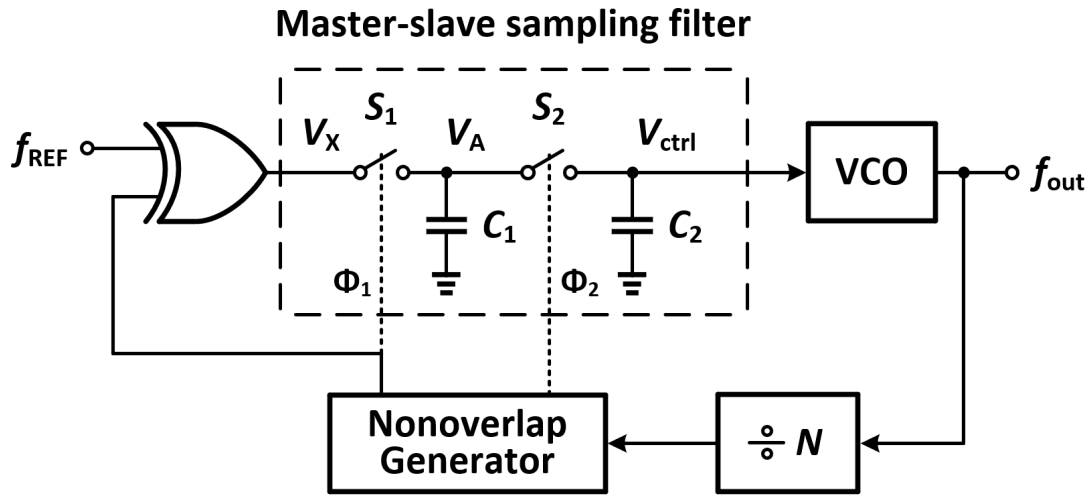


Figure 3.20: A type-I PLL using a master-slave sampling filter

The sampling clocks $\phi_{1,2}$ are produced by a nonoverlap clock generator shown in Figure 3.21 (a). Note that we choose to use transmission gates as the switches $S_{1,2}$ in the MSSF because they offer a wider input and output common-mode range for operation. Thus, the clock signal needs to be differential. First, a single-to-differential converter (S2D) converts the single-ended output of the divider to a differential sampling clock ϕ_1 and its duty cycle is about 50%. Then the signal passes through two delay cells Δt_1 and Δt_2 . The former one is related to the phase difference between ϕ_1 and ϕ_2 , while the latter one determines the pulse width of ϕ_2 . Finally, the last S2D generates the second sample clock ϕ_2 . The pulse width of ϕ_2 affects the size of the spur. If the pulse width is too large, the performance of the reference spur will deteriorate. However, if the pulse width is too

small, it may lead to insufficient charging and discharging time for the capacitor, which may prevent the PLL from locking. In this work, we design ϕ_2 pulse width ≈ 2.1 nsec, duty cycle is around 2.1%. The waveform of $\phi_{1,2}$ is illustrated in Figure 3.21 (b).

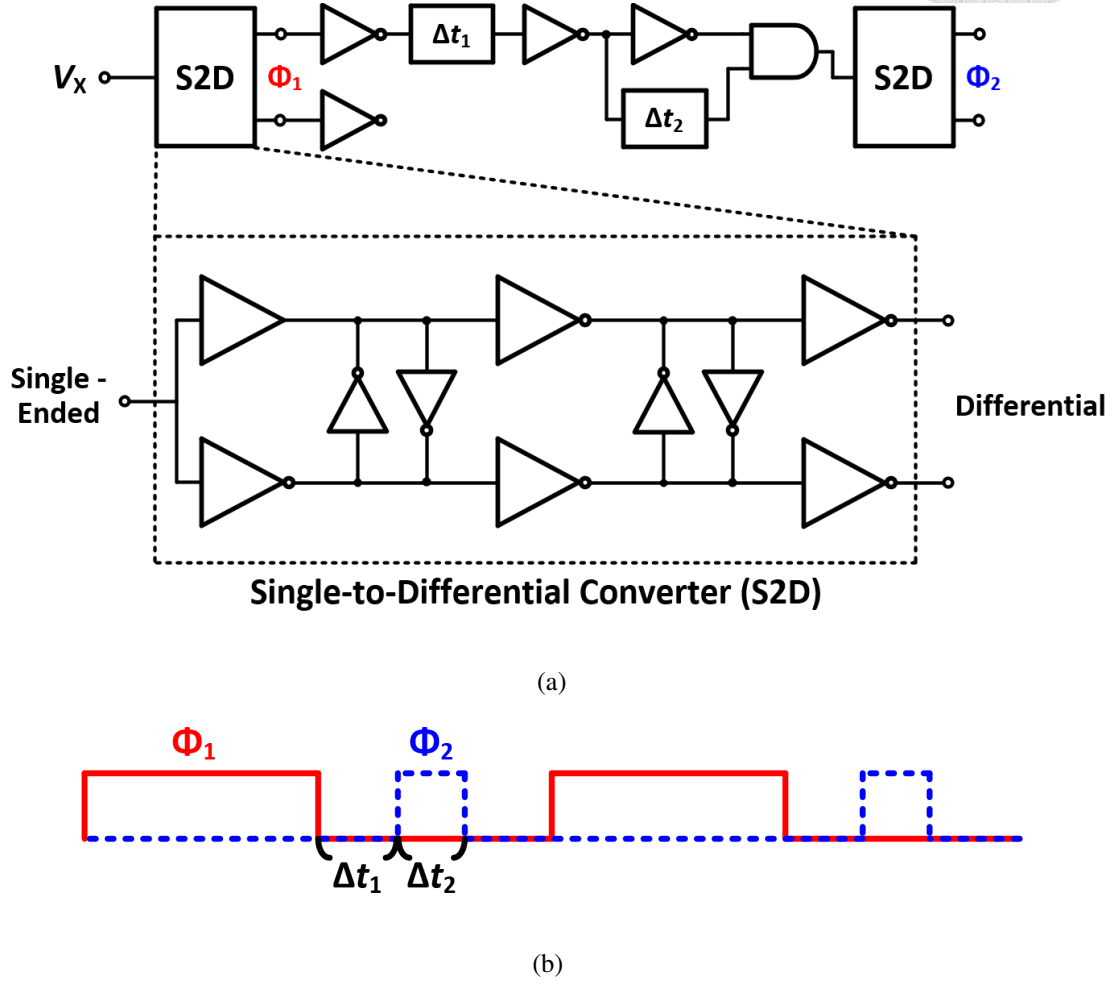


Figure 3.21: (a) Nonoverlap generator and S2D (b) Waveform of $\phi_{1,2}$

As for the analysis of MSSF, we can observe from two perspectives. For a continuous-time approximation, capacitor C_1 and two switches $S_{1,2}$ can be seen as a resistor $R_{eq} = 1/(f_{CK}C_1)$, where f_{CK} is the sampling frequency. We can formulate its transfer by

$$H(s) = \frac{1}{1 + R_{eq}C_2s} = \frac{1}{1 + \frac{C_2}{C_1f_{CK}}s} \quad (3.13)$$

Note that C_1 should be much larger than C_2 , or the voltage on V_A cannot be transmitted to V_{ctrl} .

On the other hand, we analyze the MSSF using discrete-time approximation. The MSSF generates discrete output while the input is continuous, so we can view the MSSF as a zero-order hold (ZOH) circuit if $C_1 \gg C_2$. The ZOH output can be formulated as [16]

$$Y(f) = e^{-j2\pi f T_{CK}/2} \frac{\sin \pi f T_{CK}}{\pi f T_{CK}} \sum_{n=-\infty}^{\infty} X(f - \frac{n}{T_{CK}}) \quad (3.14)$$

For the part we are interested in, $n = 0$, then

$$Y_0(f) = e^{-j2\pi f T_{CK}/2} \frac{\sin \pi f T_{CK}}{\pi f T_{CK}} \sum_{n=-\infty}^{\infty} X(f) \quad (3.15)$$

According to the equation, there will be notches at the harmonics of f_{CK} . These notches attenuate the harmonic components generated by the PD, thus suppressing the reference spur. Empirically, [1] gives a more accurate model as follows

$$H_{MSSF}(j\omega) = \frac{1}{1 + \frac{C_2}{C_1 f_{CK}} j\omega} e^{-j\pi f T_{CK}} \frac{\sin \pi f T_{CK}}{\pi f T_{CK}} \quad (3.16)$$

In this work, to demonstrate that the proposed architecture overcomes the trade-off between acquisition range and loop bandwidth, we maintain the same bandwidth as in Figure 3.1. Therefore, we choose $C_1 = 10$ pF and $C_2 = 1.3$ pF. The transient of each node in steady state is illustrated in Figure 3.22. The ripple on V_{ctrl} is suppressed to less than 0.001 V.

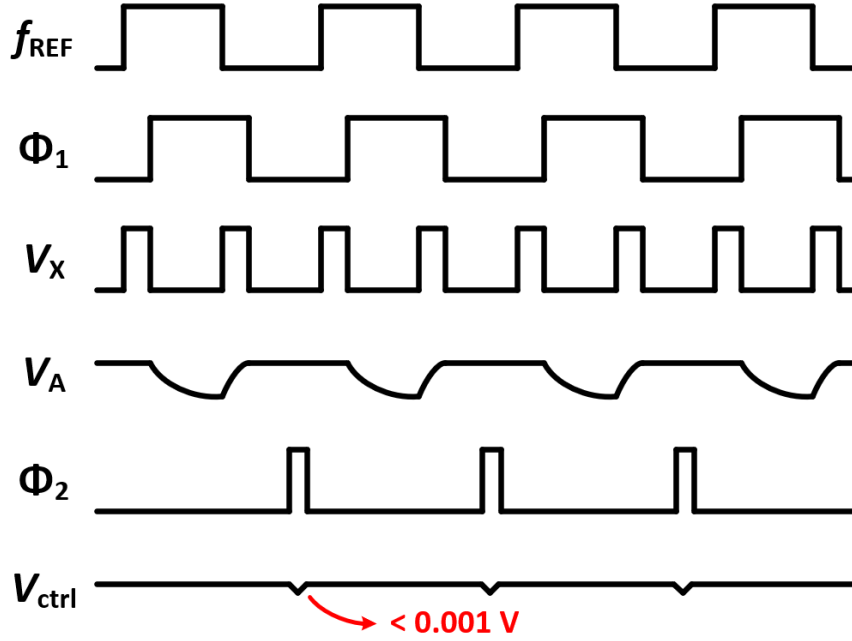


Figure 3.22: Waveform in steady state

3.4.2 Multi-Band Voltage Control Oscillator

VCO can mainly be classified into ring oscillators and LC oscillators. Generally, a ring oscillator has a wider frequency tuning range and occupies a smaller area, while an LC oscillator can generate higher output frequency and provide a lower phase noise level. To achieve the goal of a low-cost frequency synthesizer, this work plans to adopt a ring oscillator as the structure for the VCO.

Considering the acquisition range limitation of Type-1 PLL, a VCO with multiple bands is needed. For a ring oscillator, the two main factors affecting frequency are the current strength and the output loading capacitance at each stage. We utilize this characteristic and control these two variables separately: controlling the current strength to achieve band switching and adjusting the output loading capacitance at each stage for finer frequency tuning.

Therefore, we adopted the structure of a "Current Starved Ring VCO", as shown

in Figure 3.23. By configuring the current mirror matrix, we can adjust the current strength discretely to control the oscillator's output frequency band. The switches are controlled by the output codes of BSL. The more switches are turned on, the higher the output frequency of VCO. Simultaneously, mounting a varactor at the output of each stage. The capacitance is tuned continuously with V_{ctrl} in the PLL loop to achieve precise tuning effects.

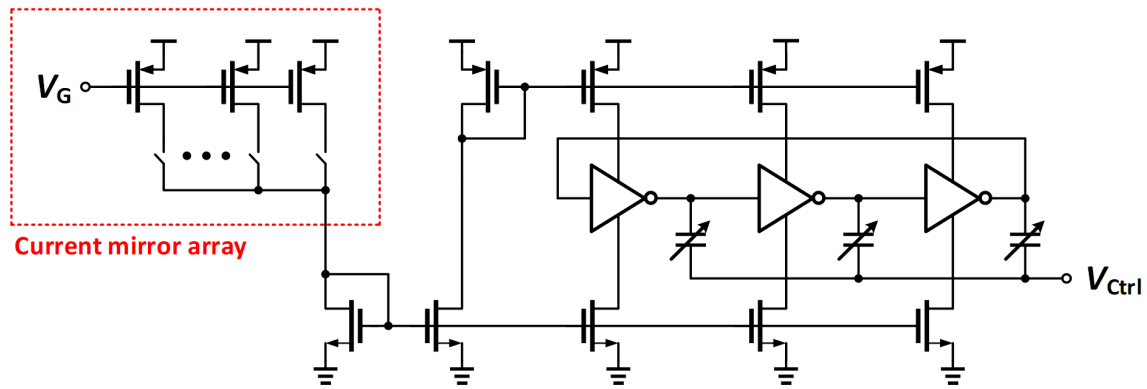


Figure 3.23: Current starved ring VCO

Figure 3.24 shows the output frequency range of VCO. A VCO with 9 bands that covers the range from 750 MHz to 2300 MHz is designed. To ensure continuous frequency generation under different process corners, each frequency band overlaps by more than 30%. Figure 3.25 shows the post-layout simulation result of VCO phase noise. The phase noise at a frequency offset of 1 MHz from the output frequency is -91.8 dBc/Hz.

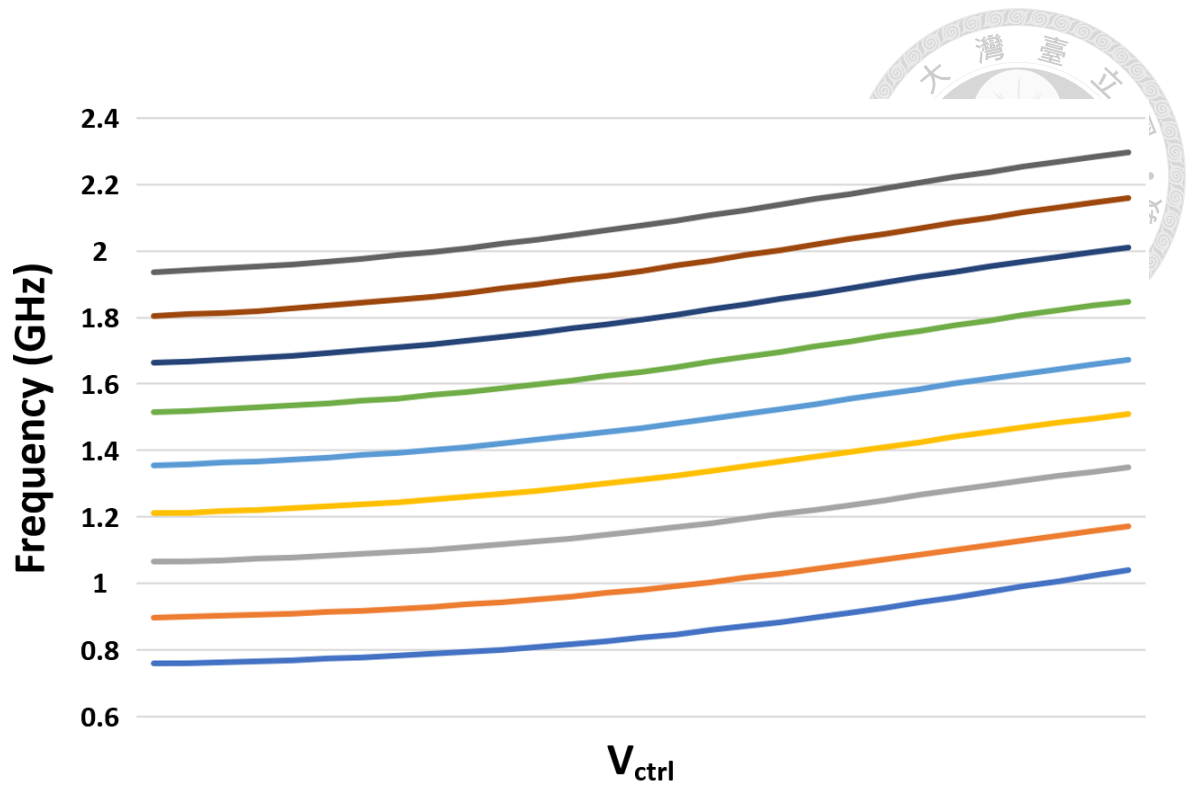


Figure 3.24: Current starved ring VCO

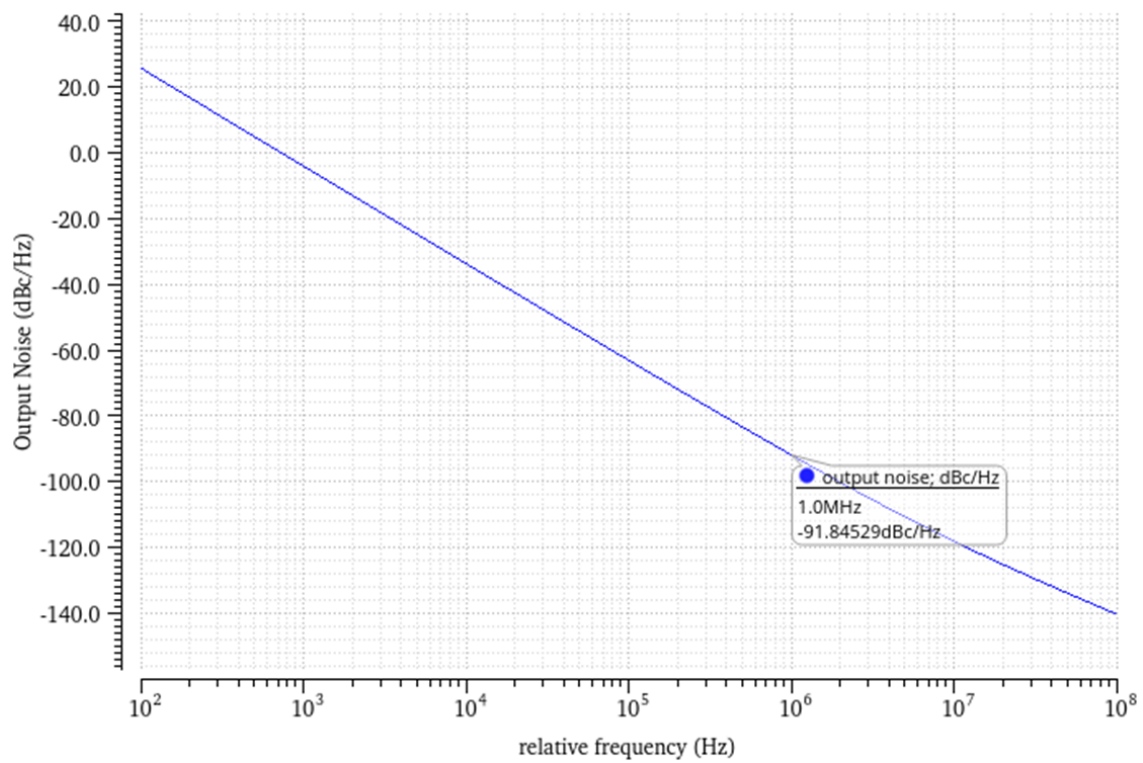
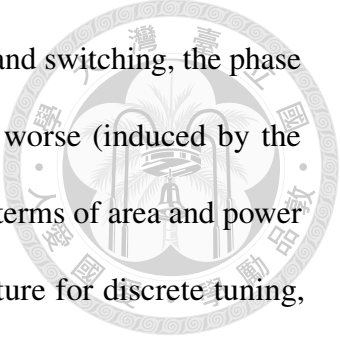


Figure 3.25: Phase noise of free-run VCO

Here we comment on the consideration that we choose this structure for discrete tuning.



Compared to the common method of using a capacitor bank for band switching, the phase noise performance of a current-starved ring VCO is relatively worse (induced by the current mirror MOS). However, this structure takes advantage in terms of area and power consumption. For example, using a capacitor bank as the structure for discrete tuning, multiple capacitors in the bank will occupy a larger area. Moreover, at the same operating frequency, since the VCO used in this work does not have capacitors connected to each output stage, the load capacitance is smaller, resulting in lower required currents. Moreover, the current-starved ring VCO also offers a wider tuning range and more consistent K_{VCO} across different frequency bands. Therefore, for the considerations of cost-saving and tuning range, we choose to apply a current starved ring VCO in this work.

3.4.3 Multi-Modulus Divider

The implementation of a multi-frequency output in a PLL can be achieved through two methods: changing the reference frequency or altering the divisor in the frequency divider. However, in practical scenarios, having multiple reference frequencies implies the need for multiple quartz oscillators, significantly increasing the cost of a frequency synthesizer. Therefore, this paper opts for the approach of using a multi-modulus divider to achieve the goal of multi-frequency output in a cost-effective manner.

We choose to use a pulse swallow divider structure, consisting of a dual-modulus prescaler, program counter, and swallow counter, as shown in Figure 3.26. At the beginning, the program counter and the swallow counter are reset. The circuit starts from the dual-modulus prescaler performing $\div(M + 1)$, it generates one pulse at A every $N + 1$ input cycles. When the swallow counter receives S cycles at node A , it switches the output state, which means switching the state of the modulus control of the prescaler. Now, the

divisor of the prescaler is changed to M . The swallow counter then continues counting cycles at node A . Until it reaches P cycles, it produces a *Reset* signal that resets the entire operation. Consequently, the overall divider generated is

$$(M + 1)S + M(P - S) = MP + S \quad (3.17)$$

Typically, by adjusting the value of S , the divisor can be varied in unity steps. Note that as explained above, the architecture requires that $P > S$.

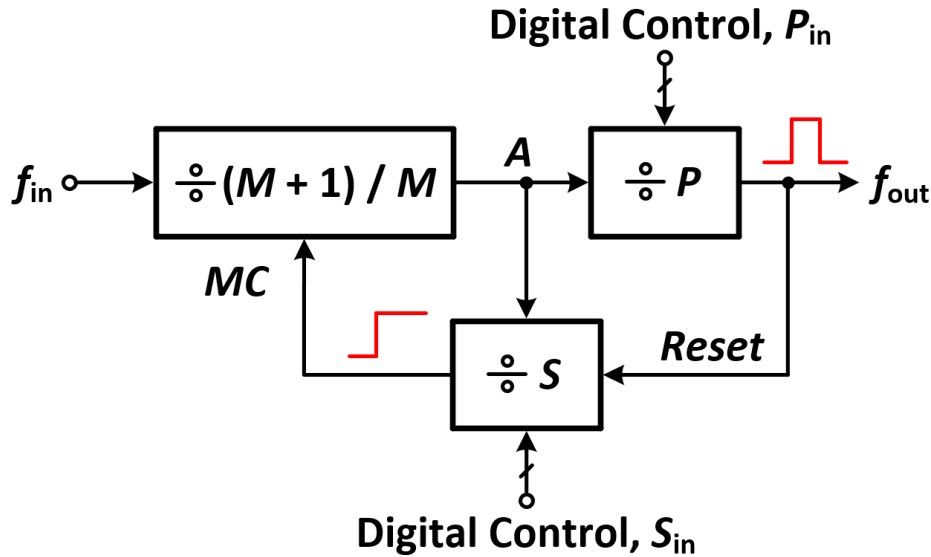


Figure 3.26: Pulse swallow divider

Recall 3.4.1, due to our adoption of XOR as the architecture for the PD in PLL, a signal of 50% duty cycle is needed. As a result, we add a divide-by-two circuit after the pulse swallow divider. Besides, we also add a divide-by-two circuit at the front of the pulse swallow divider to reduce its operating frequency to ensure its proper operation. The complete multi-modulus divider architecture of this work is illustrated in Figure 3.27. As for the divide-by-two circuit, true single-phase clocked (TSPC) DFF are exploited for power-saving, shown in Figure 3.28.

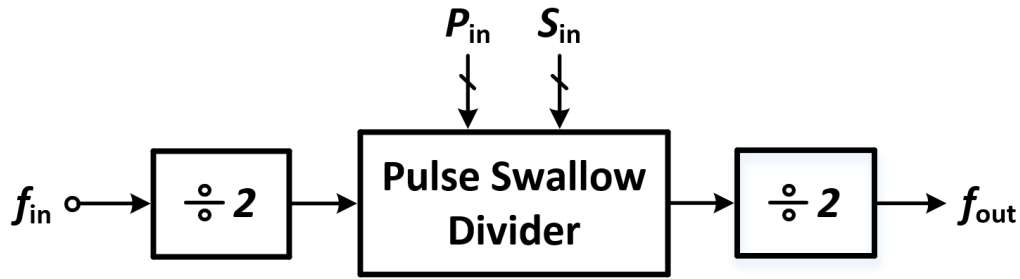


Figure 3.27: Divider structure of this work

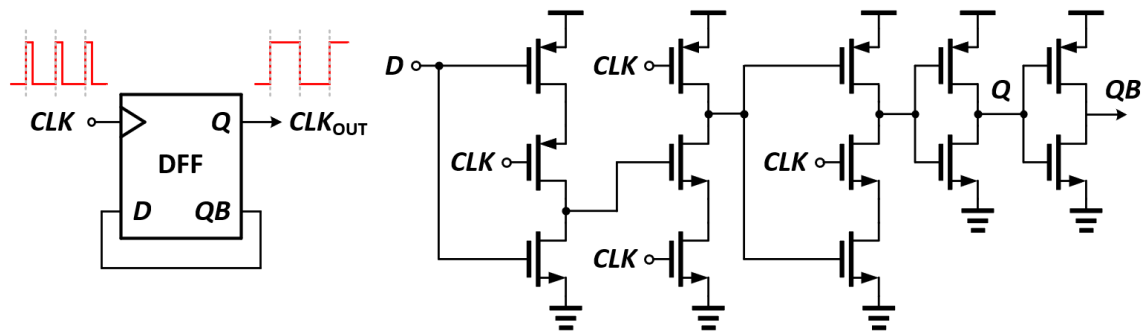


Figure 3.28: Divide-by-two circuit

In this work, we aim to generate output frequencies ranging from 840 MHz to 2240 MHz with a 10 MHz reference. Since we have two divide-by-two circuits in the entire divider chain, the output frequency step becomes 40 MHz. The modulus of the pulse swallow divider ranges from 21 to 56, producing 36 different divisors in total.

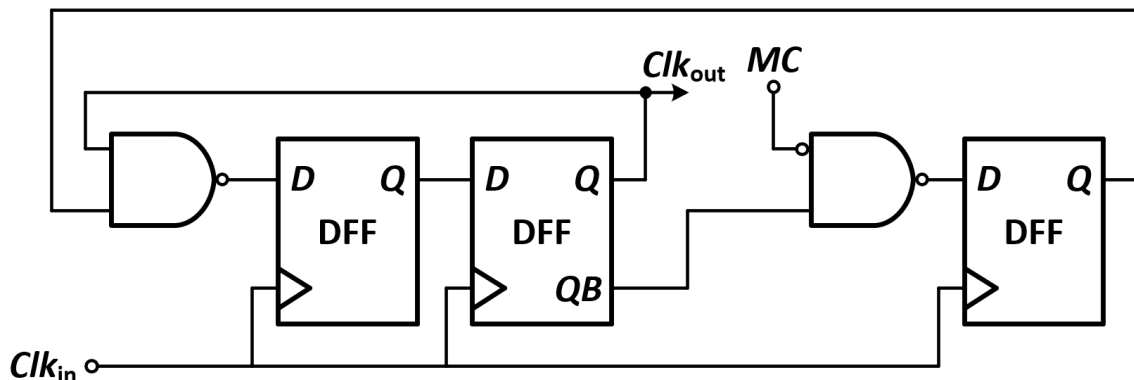
As we just mentioned, the pulse swallow divider operates under the condition where $P > S$. However, it is impossible to satisfy this rule simultaneously in situations where

$$MP + S_{36} = 56$$

Let $S_1 = 1, S_{36} = 36$

Even though S and M are minimized as much as possible, it's still possible for S to exceed P in certain cases. To fit the rule, we make the program counter programmable as well. The design parameters are $M = 4$, $P = 5 \sim 13$, and $S = 1 \sim 4$. In the following paragraphs, we will discuss the implementation of each block in the pulse swallow divider, which includes the dual-modulus prescaler, program counter, and swallow counter.

- The schematic of the dual-modulus prescaler is shown in Figure 3.29. When MC is low, the prescaler functions as a $\div 5$ circuit. Conversely, when MC is high, the prescaler operates as a $\div 4$ circuit. The waveform is shown in Figure 3.30.



53

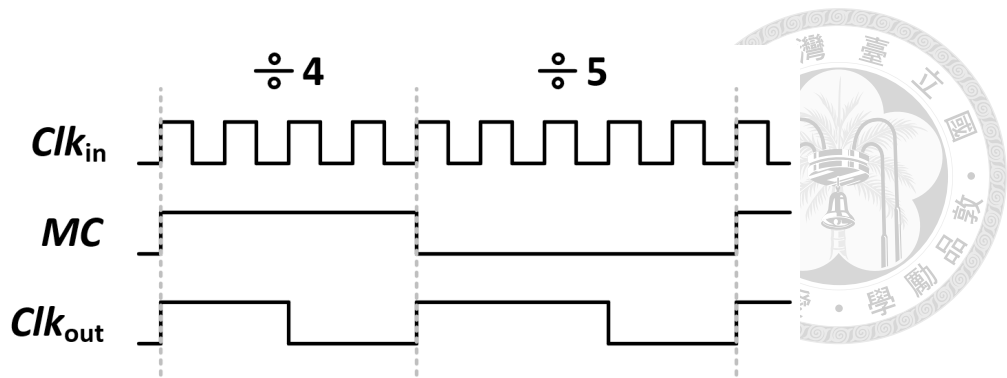


Figure 3.30: Waveform of the dual-modulus prescaler

- Program counter:

The program counter generates a pulse to reset other blocks after receiving P input cycles. Besides, to further expand the adjustable range of the divisor, the variable P is designed to be programmable from $5 \sim 13$, controlled by a set of control signals P_{in} .

Figure 3.31 is the block diagram of program counter. The counter at the first stage calculates input cycles and outputs the result X to the comparator. When $X = P_{in}$, the comparator outputs *High*. However, directly using this signal as the *Reset* signal to reset the counter would cause it to reset immediately after rising, resulting in a too-short pulse width. This could lead to incomplete discharge of the DFF in the counter and cause logic errors. Therefore, after the comparator, a falling edge-triggered DFF is connected as a register to ensure that the reset pulse width is at least one input cycle long, thus ensuring the correctness of the circuit logic.

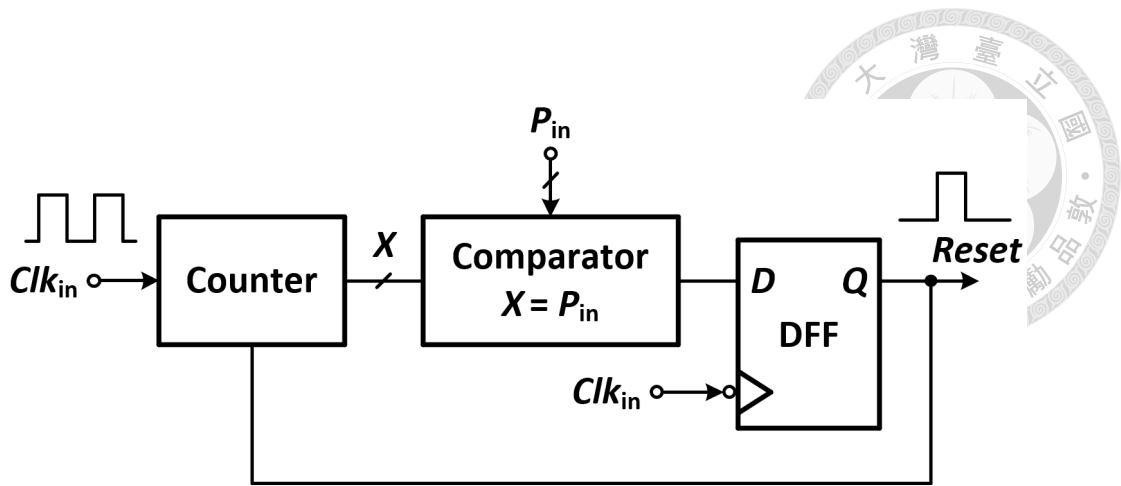


Figure 3.31: Programmable program counter

Note that because of the register at the final stage, the program counter will have one more input cycle in total, resulting in the actual value of P being $P = P_{in} + 1$.

The waveform is illustrated in Figure 3.32 when $P_{in} = 5$.

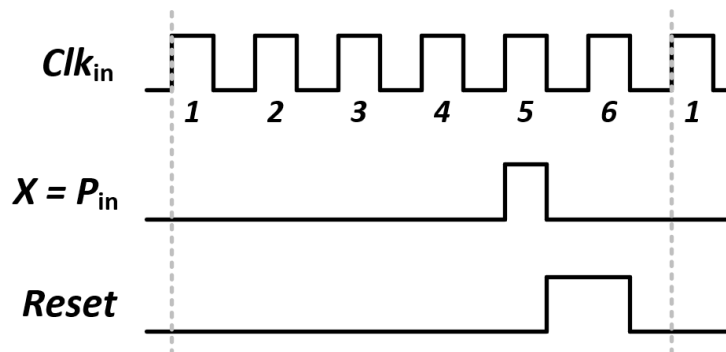


Figure 3.32: Waveview of program counter when $P_{in} = 5$

- Swallow counter:

The swallow counter switched its output stage MC to change the modulus of the prescaler after receiving S input cycles. The variable S is designed to be programmable from 1 ~ 4, controlled by a set of control signals S_{in} .

Figure 3.33 is the block diagram of a swallow counter. The counter at the first stage calculates input cycles and outputs the result Y to the comparator. When $Y = S_{in}$,

the comparator outputs *High*. This value is then stored in a latch causing *MC* to lock at *HIGH* as well. Until the *Reset* signal from the program counter is received, *MC* will be reset to 0.

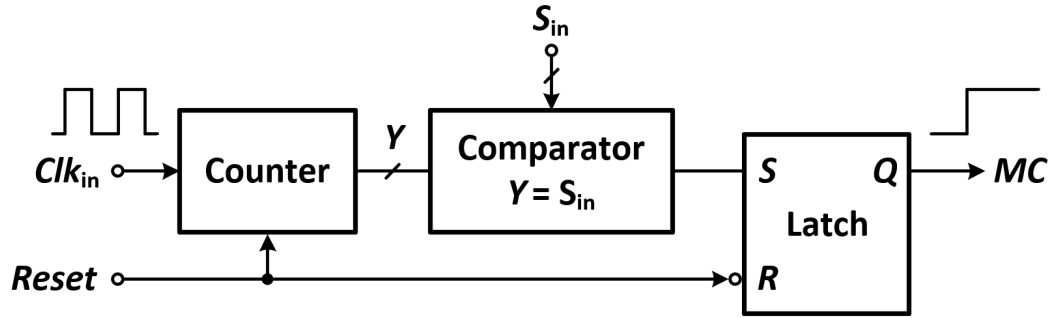


Figure 3.33: Programmable swallow counter

The waveform of the swallow counter is illustrated in Figure 3.34. When the *Reset* signal is *HIGH*, the counter must output 0, indicating that its output *Y* starts from 0. Consequently, when $Y = S_{in}$, it implies that the counter has already passed $S_{in} + 1$ input cycles, resulting in the actual value of *S* being $S = S_{in} + 1$.

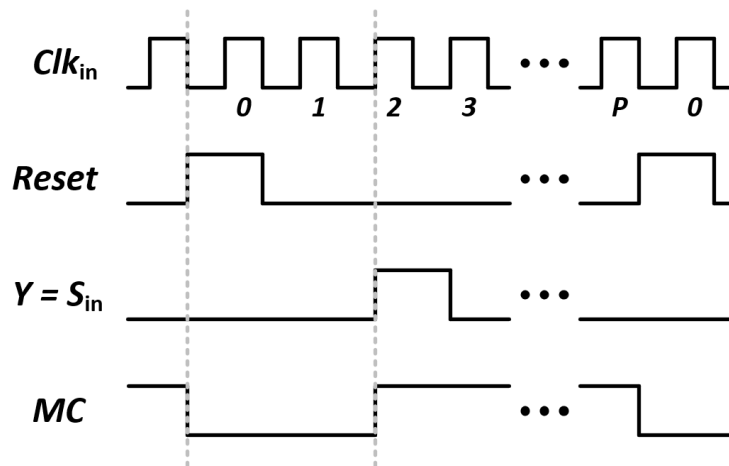


Figure 3.34: Waveview of swallow counter when $S_{in} = 2$

The modulus of the pulse swallow divider is calculated as $M(P_{in} + 1) + (S_{in} + 1)$, where $M = 4$, P_{in} is a 4-bit digital control signal and S_{in} is a 2-bit digital control signal.

It provides a total of 48 consecutive integer dividers ranging from 21 to 68. In post-simulation, it can operate at frequencies up to 2 GHz. Figure 3.35 is a transient simulation of the pulse swallow divider, the input frequency is set to be 1.3 GHz, $P_{in} = 5$ and $S_{in} = 1$. The output frequency is 40 MHz.

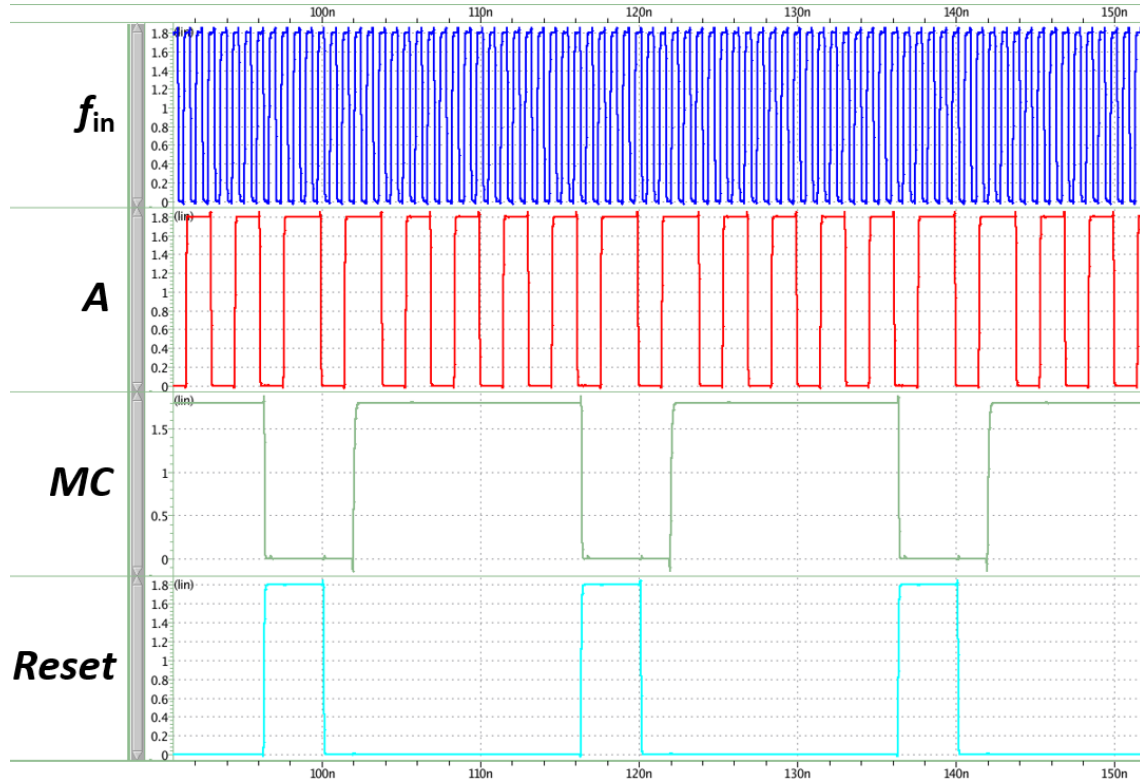


Figure 3.35: Transient simulation of pulse swallow divider

With two additional divide-by-two circuits before and after the pulse swallow divider, the overall divider can generate 84, 88, 92... up to 272, with intervals of 4, totaling 48 different dividers. The divisor of PLL can be formulated as

$$N = 4 \times (M(P_{in} + 1) + (S_{in} + 1)) \quad (3.18)$$

3.4.4 Band-Selecting Loop

In 3.3, we introduced a novel band-selecting mechanism and its implementation. Here, we will provide some additional details related to other blocks in the main loop.

In the proposed BSL, we need the value N , which is the divisor of PLL, to define the upper bound and the lower bound of comparators. However, as mentioned in 3.4.3, what we input are only P_{in} and S_{in} . Divisor is actually calculated through (3.18). Hence, the circuits are required to compute, or decode, inputs P_{in} and S_{in} to obtain N .

Let's recall (3.5) and (3.18), Summarized as below:

$$\begin{cases} N - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < N + \frac{R}{f_{REF}} \\ N = 4 \times (M(P_{in} + 1) + (S_{in} + 1)) \end{cases} \quad (3.19)$$

Dividing both equations in (3.19) by 4, we get:

$$\begin{cases} \frac{N}{4} - \frac{R}{4 \times f_{REF}} < \frac{f_{out}/4}{f_{REF}} < \frac{N}{4} + \frac{R}{4 \times f_{REF}} \\ \frac{N}{4} = M(P_{in} + 1) + (S_{in} + 1) \end{cases} \quad (3.20)$$

The parameter $\frac{f_{out}/4}{f_{REF}}$ indicates that the output of the VCO needs to be divided by four before being fed into the BSL. The $\div 4$ circuit can be realized by cascading two $\div 2$ circuits as Figure 3.28.

Besides, $N/4$ is the modulus of pulse swallow divider, denoted as N' . We need a circuit to generate the value N' on chip with inputs P_{in} and S_{in} . For convenience, we call this circuit a decoder.

The decoder is composed of three adders. The first one computes $S_{in} + 1$, while the second one computes $P_{in} + 1$. Next, to multiply $P_{in} + 1$ by M , traditional multiplication

in logic circuits could be complex. However, for division by 2^x in binary code, it simply requires shifting the bits to the left for x positions. This is why we design $M = 4$ in 3.4.3 for the pulse swallow divider. Since $4 = 2^2$, shifting $P_{in} + 1$ two bits to the left yields the result of $(P_{in} + 1) \times 4$. Consequently, we only need to add this shifted binary string to the previously obtained $S_{in} + 1$ using another adder to derive the binary code of N' . Figure 3.36 is the diagram of the decoder, demonstrating the scenario when $P_{in} = 5$ and $S_{in} = 1$.

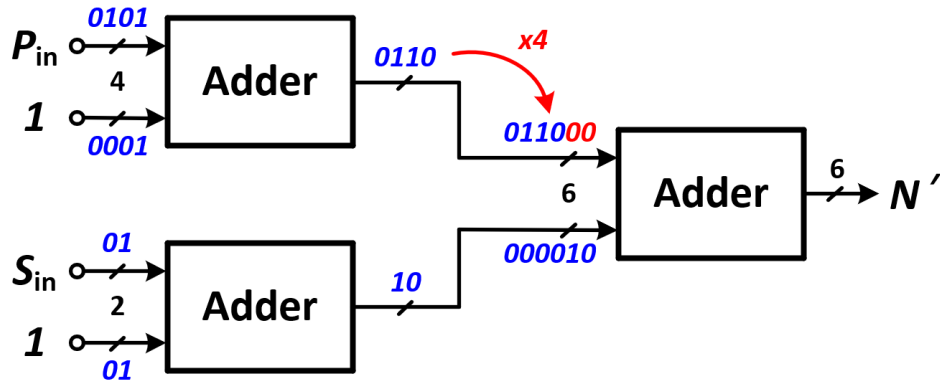


Figure 3.36: Decoder when $P_{in} = 5$ and $S_{in} = 1$

Let's move to another part. We know that BSL generates a set of 8-bits thermometer codes to control the output band of VCO. However, it controls the connection between VCO and the main loop as well.

When BSL is activated, the VCO is disconnected from the main loop, and V_{ctrl} is connected to a fixed voltage, V_c . So there will be two switches control where V_{ctrl} connects to: one switch is between LF and VCO, and the other is between VCO and V_c . These switches are controlled by a pair of complementary signals. Referring back to 3.3.4, an XNOR in BSL senses C_1C_0 and then outputs a *Power* signal, indicating whether the band-selecting process is ongoing. Here we can utilize *Power* and its complement $\overline{Power}$ to control the switches, as illustrated in Figure 3.37. When the BSL is in progress,

V_{ctrl} is connected to V_c . Once the band-selecting process concludes, $Power$ returns to 0, prompting the reconnection of the VCO to the main loop while disconnecting V_c . Note that V_c will be set to the corresponding V_{ctrl} value at the midpoint of the VCO tuning range frequency.

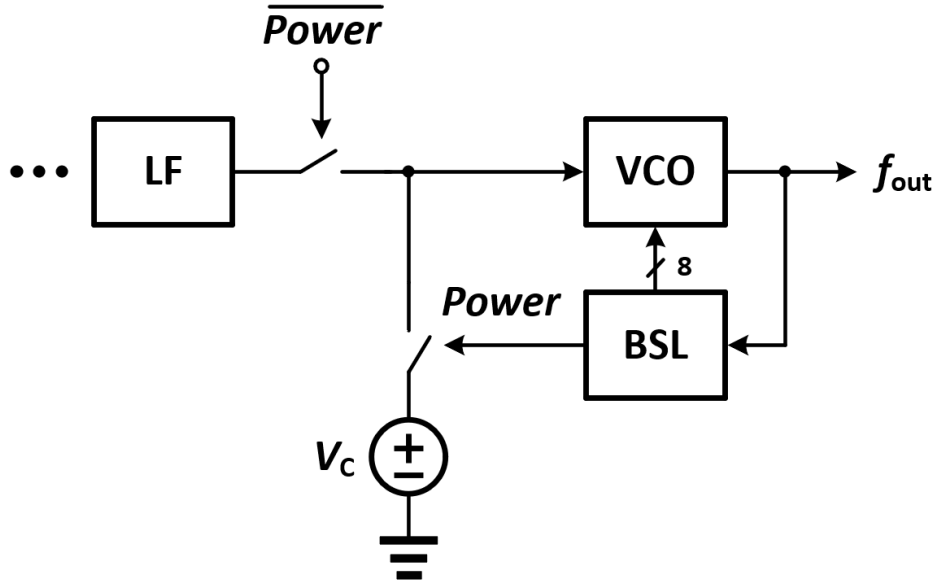


Figure 3.37: Connection between VCO and the main loop

3.5 Summary

The proposed wide frequency output range type-I PLL with multiple modulus is presented in this chapter. With an auxiliary BSL, the loop can be prevented from unlocking because of exceeding the acquisition range. The overall architecture is clarified first. Then we present the proposed band-selecting mechanism, including the concept, operation flow, and implementation. Furthermore, the circuit design of each building block is detailed. Parameters selection between each block was also mentioned. The experimental results will be demonstrated in the next chapter.



Chapter 4

Experimental Results

4.1 Chip Photo

The proposed wide-range type-I PLL with a band-selecting loop is fabricated in the TSMC 1P6M 180-nm CMOS process. The chip photo is shown in Figure 4.1. The active area is $0.35 \times 0.35 \text{ mm}^2$. The chip comprises several power domains, including those for the VCO, BSL, buffers, and other blocks. These domains are isolated to prevent noise and signal coupling via power lines. Additionally, there are 6 pads for modulus control, 2 for swallow counter parameter S , and 4 for program counter parameter P .

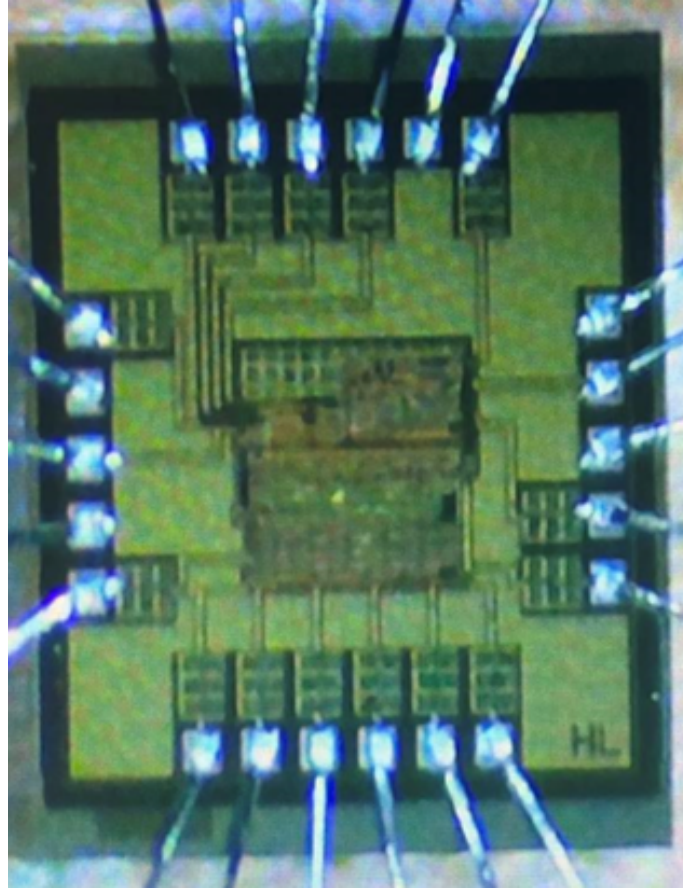


Figure 4.1: Chip photo

4.2 Measurement Setup

Figure 4.2 illustrates the diagram of the measurement setup. The chip is wired and bonded directly to a printed circuit board (PCB). The capacitance values of Surface Mount Device (SMD) capacitors from the inner to outer layers are 1 μF , 0.01 μF , and 100 pF. They are soldered from the DC power supplies to the ground to bypass high-frequency noises effectively. SubMiniature version A (SMA) connectors establish connections: one for the reference clock from the signal generator to the chip and another for the PLL output clock from the chip to the signal analyzer. Power supplies (Agilent E3649A) provide V_{DD} DC voltages for each power domain. A signal generator (KEYSIGHT 33600A) produces a

10-MHz clock as the reference signal f_{REF} of the PLL. The output clock signal f_{out} is linked to the signal analyzer (Anritsu MS2690A), which facilitates measurements of the spectrum and phase noise.

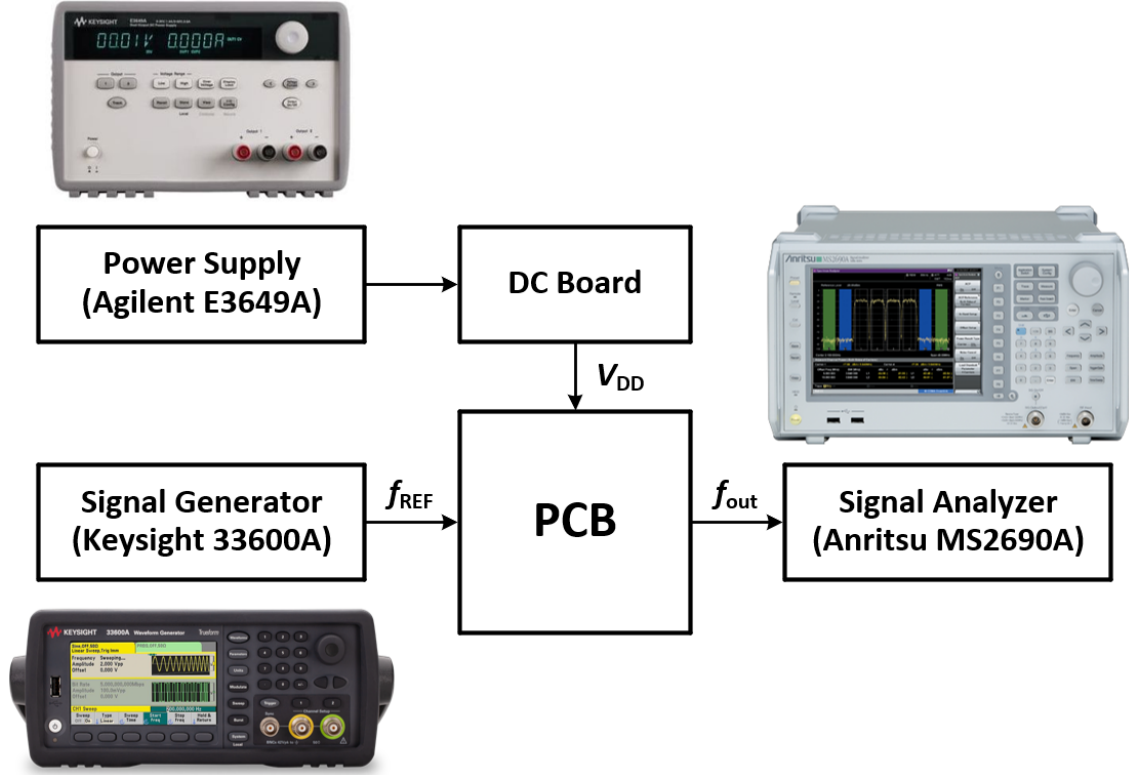
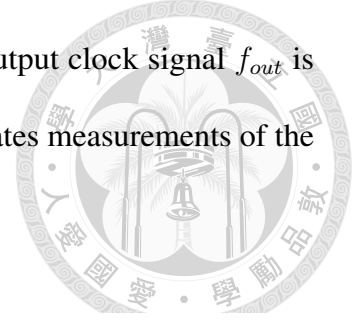


Figure 4.2: Measurement setup

4.3 Experiment Result

The simulated transient response of the proposed PLL is shown in Figure 4.3. In the beginning, the divisor is set to 100 and the target frequency is thus 1000 MHz. At $t = t_1$, switch the divisor to 200. As soon as the divisor changes, the BSL turns on. During this process, the VCO is disconnected from the main loop, and V_{ctrl} is set to a constant value. At $t = t_2$, the output band falls within the acquisition range of the target frequency, so the power of BSL turns off and reconnects the VCO to the main loop. Within a few microseconds, V_{ctrl} settle to a proper value that locks the PLL to 2000 MHz.

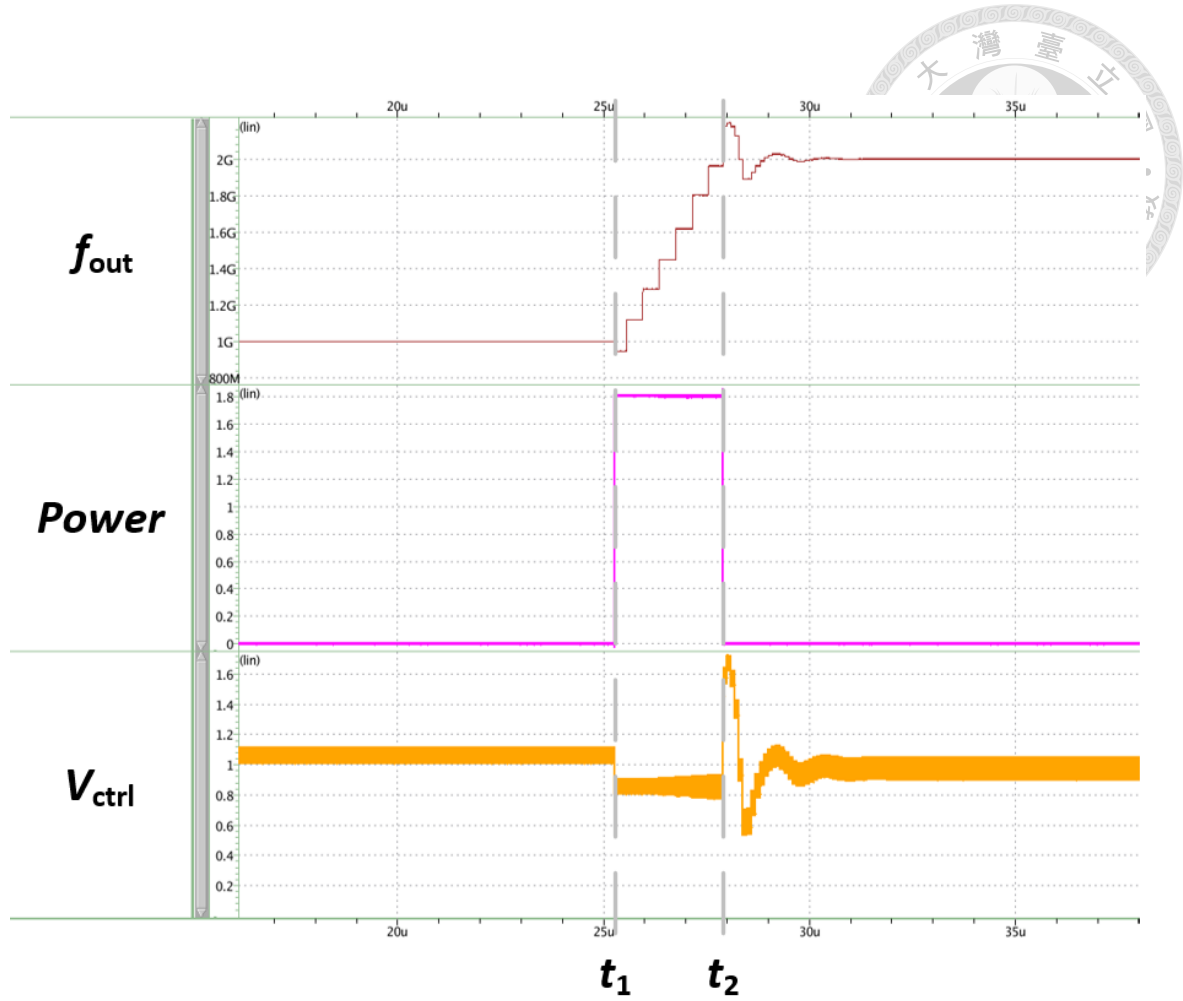


Figure 4.3: Transient response simulation of the system

The proposed PLL can operate from 840 MHz to 2240 MHz. Figure 4.4 and Figure 4.5 are the measured spectra of the proposed type-I PLL with the BSL at the lowest and the highest frequency, respectively. The reference spur is -53 dBc at 840 MHz and -50 dBc at 2240 MHz. If the BSL is not activated, unlocking occurs when the frequency switch between the front and back exceeds the acquisition range. Figure 4.6 shows the spectrum when unlocking occurs.

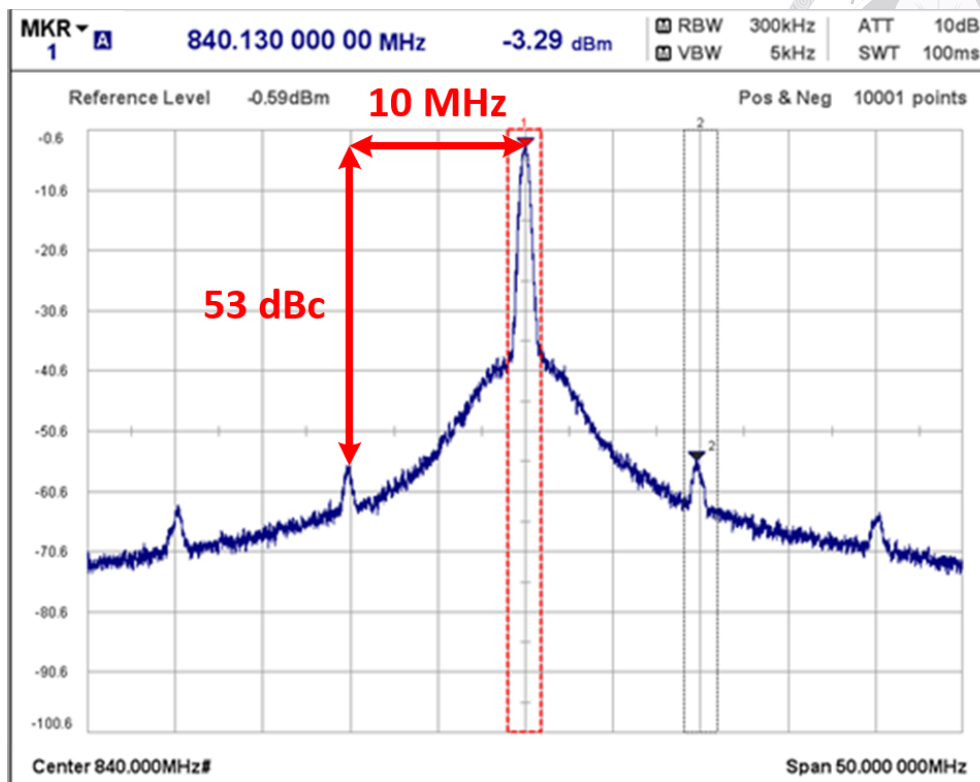


Figure 4.4: Measured spectrum of the proposed type-I PLL at 840 MHz

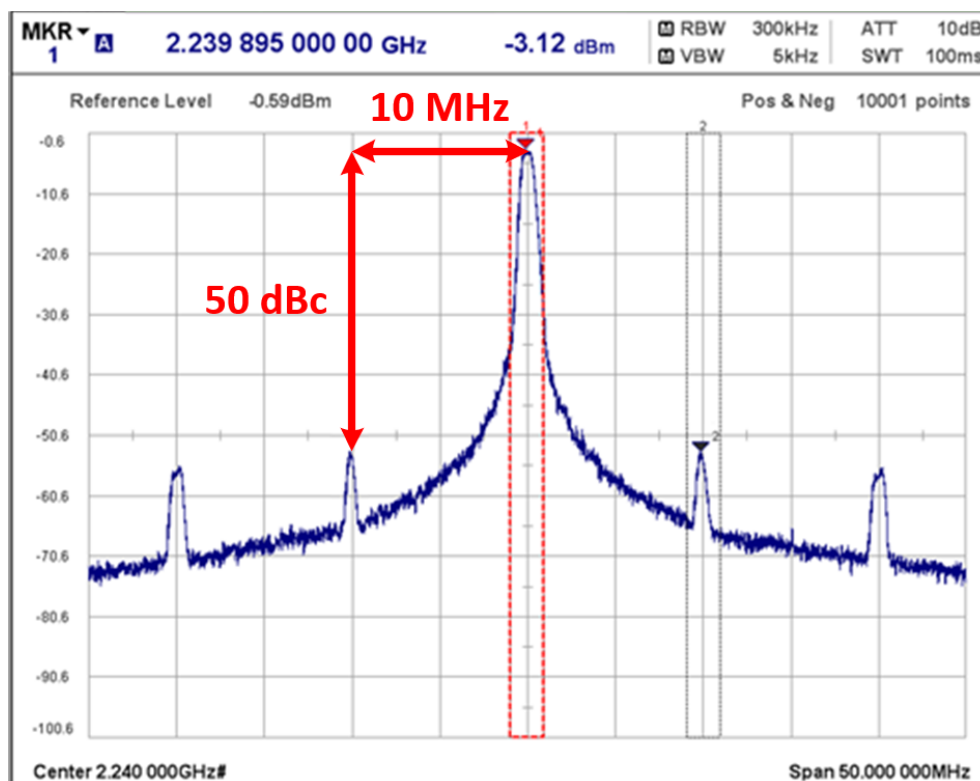


Figure 4.5: Measured spectrum of the proposed type-I PLL at 2240 MHz

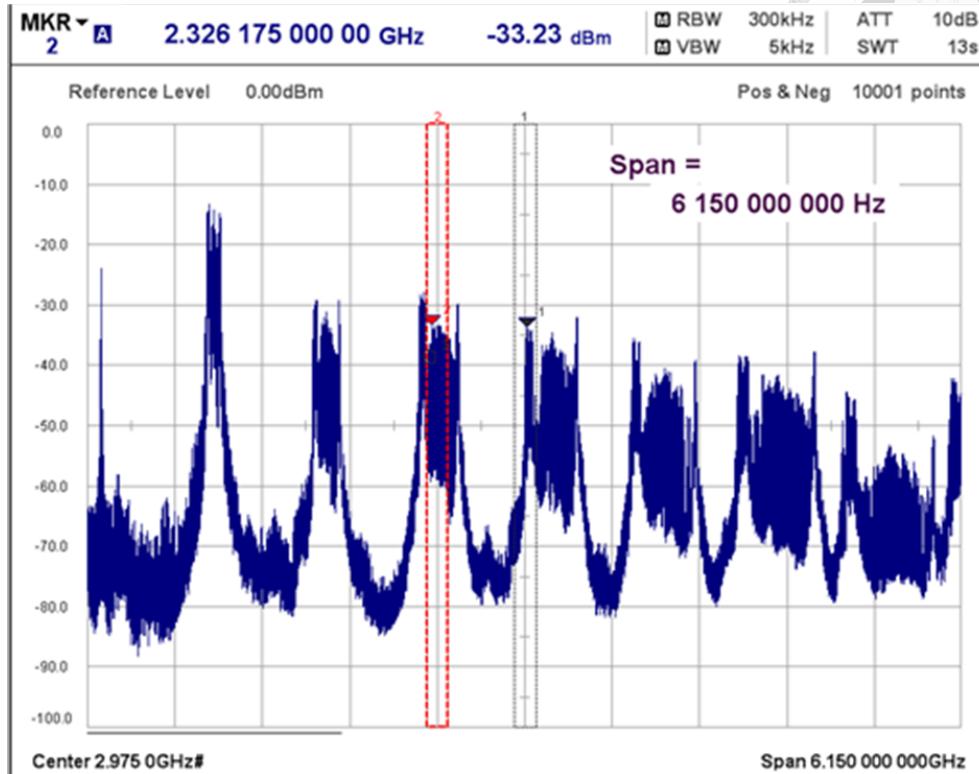


Figure 4.6: Measured spectrum when the PLL is unlocked

The measured phase noise of the proposed type-I PLL at 840 MHz and 2240 MHz are displayed in Figure 4.7 and Figure 4.8. The measured phase noise at the 1 MHz offset is -92.45 dBc/Hz and -91.70 dBc/Hz, respectively. Additionally, to assess the noise contribution of the BSL, we disable the BSL in the experiment conducted in the range of 840 MHz to 920 MHz to examine the phase noise difference. The measurement results demonstrate that enabling the BSL does not introduce additional noise to the output.

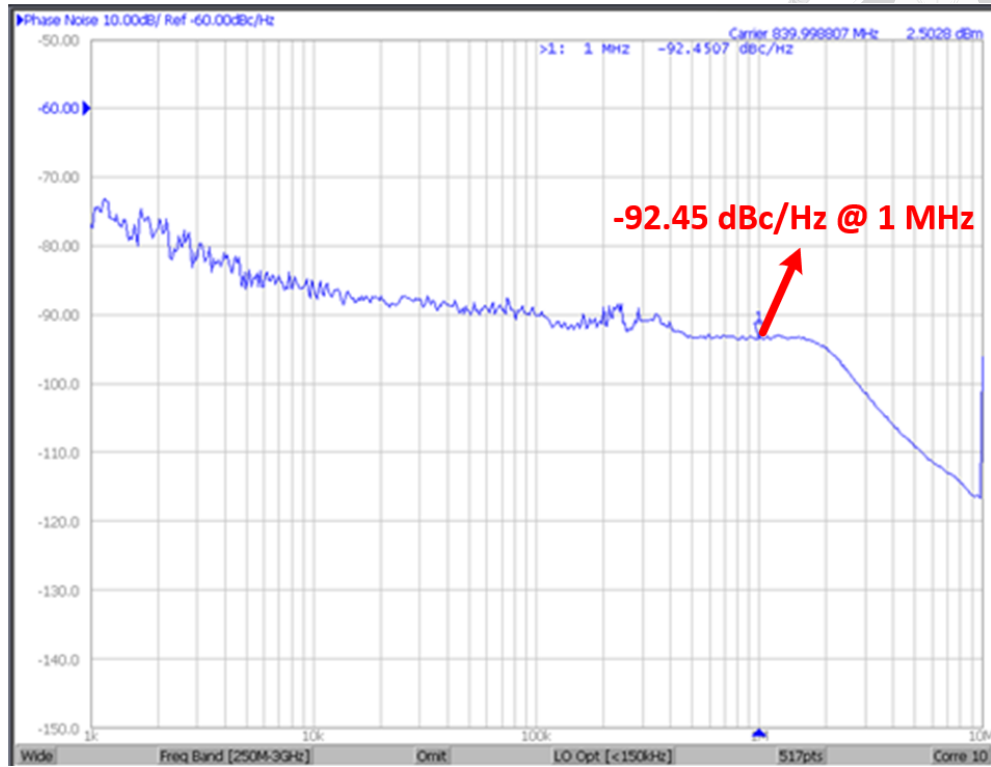


Figure 4.7: Measured phase noise of the proposed type-I PLL at 840 MHz

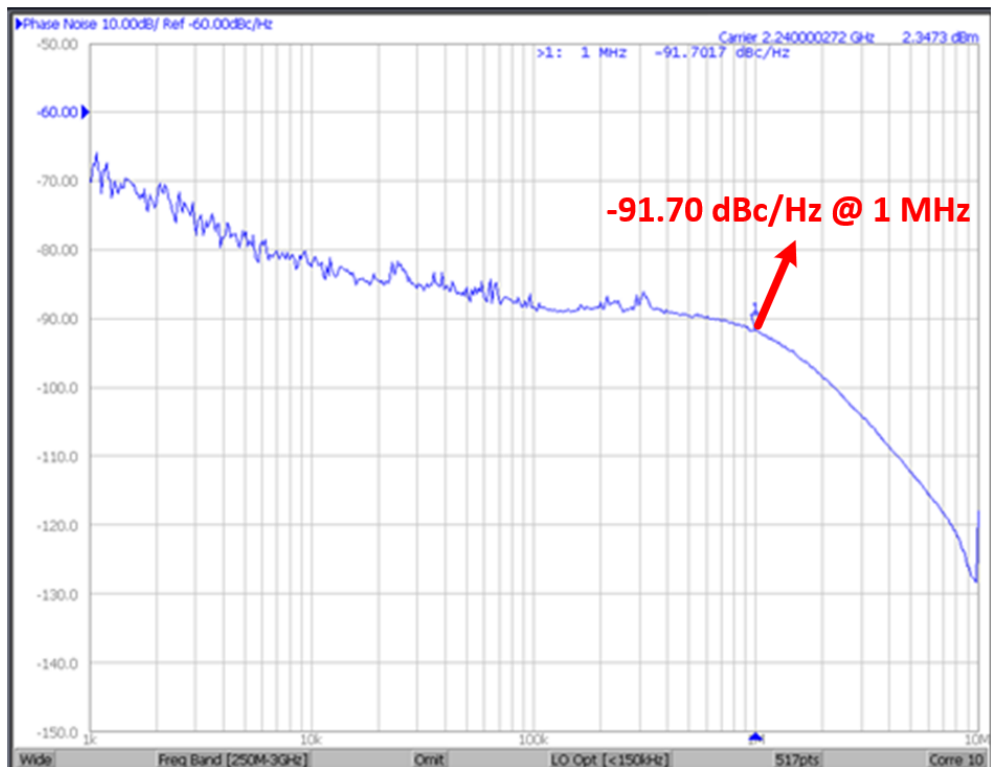


Figure 4.8: Measured phase noise of the proposed type-I PLL at 2240 MHz

4.4 Comparison

Table 4.1 summarizes and compares the performance with recently-published PLLs [17][1][18][19][20]. To overcome the limited acquisition range, [17] used auxiliary LF paths with a quasi-type-II/I implementation. However, this approach incurred significant costs in terms of area and power consumption. [1] introduced the MSSF architecture referenced in this paper. It is noteworthy that this paper additionally included a harmonic trap circuit to further reduce the reference spur. When the harmonic trap circuit is not activated, the reference spur is -47 dBc. In comparison, this paper significantly enhances the tuning range without increasing the reference spur. This proves that we resolve the trade-off mentioned earlier. [18] improved the acquisition range by increasing the forward loop gain. Despite similar area requirements, this paper still holds a substantial advantage in tuning range. Both [19] and [20] adopted Type-II architectures. However, this paper clearly achieves a larger tuning range in a more cost-effective manner, including area and power consumption.

4.5 Summary

This chapter demonstrates the experiment results of the presented wide-range type-I PLL, including the transient response, output clock spectrum, and phase noise. The proposed architecture provides a total of 48 output frequencies ranging from 840 MHz to 2240 MHz. With the BSL, this work has surpassed the limitations of the acquisition range in type-I PLLs, achieving approximately 90.9% tuning range without degrading the reference spur. This work provides a significantly expanded frequency operating range compared to other studies in the field.



Table 4.1: Comparison table

—	[17] RFIC'12	[1] JSSC'16	[18] TCAS-I'19	[19] JSSC'15	[20] ISSCC'11	This Work
Technology (nm)	130	45	130	65	65	180
Supply (V)	1.2	1.0	1.2	1.0	1.2	1.8
OSC	LC	Ring	Ring	Ring	Ring	Ring
Reference (MHz)	21	22.6	19.2	50	55	10
Output (GHz)	2.7 ~ 5.4	2 ~ 3	2.2 ~ 2.8	4.25 ~ 4.75	1.0 ~ 1.5	0.84 ~ 2.24
TR (%)	65%	40%	24%	11%	40%	90%
PN @ 1 MHz (dBc/Hz)	-94	-113.8	-114.1	-103.8	-119	-91.7
Ref. Spur (dBc)	-55	-47 / -65	-65	-60	-42	-50
Power (mW)	19.5	4	6.8	11.6	51.6	7.2
Active Area (mm ²)	1.5	0.015	0.12	0.48	0.12	0.13





Chapter 5

Modified Design

In this chapter, we will review this work and make improvements to certain parts that can be optimized. The modified chip has been completed with layout and was taped out in February of this year. Layout diagrams and post-simulation results will be presented later.

5.1 Work Review

Reviewing this work, one obvious issue is that the area occupied by the BSL is quite large, about $0.2 \times 0.35 \text{ mm}^2$. The layout of the core area is depicted in Figure 5.1, illustrating the arrangement of each block within the BSL. Recall that the decoder is responsible for producing the N values for the two adders to generate the upper bound and lower bounds. We can notice that the decoder and two adders together occupy approximately half of the area within the BSL. In the following paragraph, we will try to reduce the area occupied by these surrounding circuits. Another block that occupies a large area is the pulse swallow divider. We will try to fix this as well.

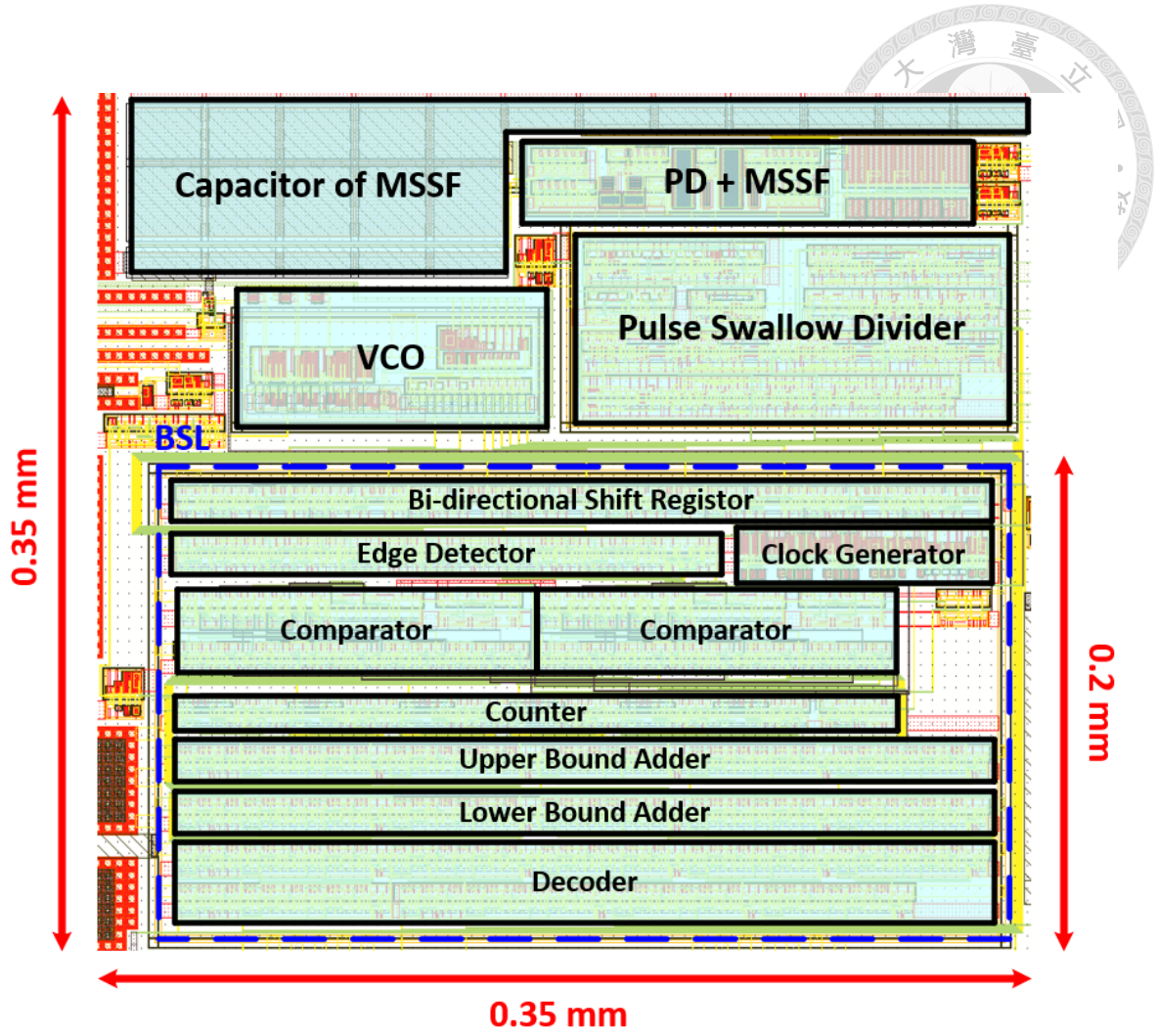


Figure 5.1: The arrangement of BSL in layout

Recall the equation (3.19), we rewrite it below

$$\left\{ \begin{array}{l} N - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < N + \frac{R}{f_{REF}} \end{array} \right. \quad (5.1)$$

$$\left\{ \begin{array}{l} N = 4 \times (M(P_{in} + 1) + (S_{in} + 1)) \end{array} \right. \quad (5.2)$$

First of all, the equation (5.2) implies that to generate N with input P_{in} and S_{in} , we need at least three adders to construct a decoder, which results in a significant occupation of the decoder. The idea is that if we can simplify the generation of the divisor to have only one input, P_{in} , then the new equation of N would be like $N = K \times (P_{in} + 1)$, and we should be able to reduce the number of adders required for the decoder. Now, we can formulate

the equation as

$$\begin{cases} N - \frac{R}{f_{REF}} < \frac{f_{out}}{f_{REF}} < N + \frac{R}{f_{REF}} \\ N = K \times (P_{in} + 1) \end{cases} \quad (5.3)$$



Divide both equations by K we get

$$\begin{cases} N/K - \frac{R}{f_{REF}K} < \frac{f_{out}/K}{f_{REF}} < N/K + \frac{R}{f_{REF}K} \\ N/K = P_{in} + 1 \end{cases} \quad (5.5)$$

Substitute $N/K = P_{in} + 1$ into the equation (5.5), we have

$$P_{in} + 1 - \frac{R}{f_{REF}K} < \frac{f_{out}/K}{f_{REF}} < P_{in} + 1 + \frac{R}{f_{REF}K} \quad (5.7)$$

In equation (5.7), we've discovered something interesting: if $\frac{R}{f_{REF}K} = 1$, then the equation can be optimized to the simplest

$$P_{in} < \frac{f_{out}/K}{f_{REF}} < P_{in} + 2 \quad (5.8)$$

In this case, P_{in} directly represents the lower bound of the acceptable range. We only need one adder to compute $P_{in} + 2$ to obtain the upper bound, saving the entire decoder and one adder's worth of space. As for $\frac{f_{out}/K}{f_{REF}}$, simply adding a divide-by- K circuit before f_{out} enters the BSL will allow us to obtain the result of it from the output of the counter.

Additionally, since now $N = K \times (P_{in} + 1)$, we can simplify the divider circuit to just a programmable program counter cascade with a divide-by- K circuit. The area of the divider can also be reduced.

In short conclusion, by appropriately selecting parameters f_{REF} and K , as well as

simplifying the divisor's equation, we can optimize our area planning of the BSL and the divider.



5.2 Adjusted Block

To make $\frac{R}{f_{REF}K} = 1$, we redesign the frequency planning. The new value for $f_{REF} = 40$ MHz, and K is set to be 4, as R remains the same at 160 MHz. We will provide an explanation of why we chose these parameters.

The adjusted divider is shown in Figure 5.2. Since the maximum operating frequency of the program counter in the post-simulation is 1200 MHz, while we designed the PLL to have a maximum output frequency of 2240 MHz, it's necessary to include a $\div 2$ circuit in the first stage of the divider. This ensures the correctness of circuit operation by reducing the frequency f_{out} before it enters the program counter. Besides, the $\div 2$ circuit at the final stage is designed to generate a 50% duty cycle output signal for use by the XOR-based PD. These two $\div 2$ circuits effectively provide a $\div 4$ effect. Thus, we design $K = 4$ so that we eliminate the need for additional circuitry to achieve divide-by- K . We minimize the required circuitry as much as possible while ensuring that each block operates correctly, thereby reducing chip area.

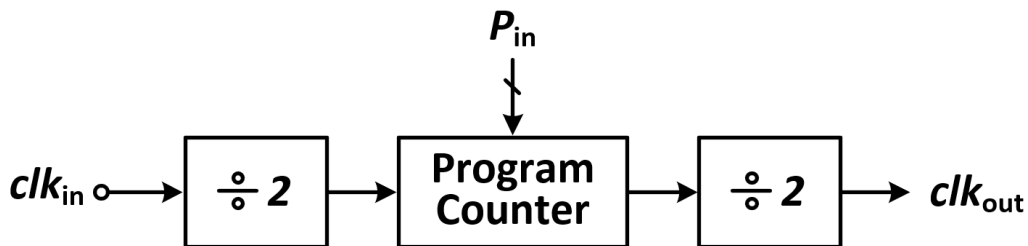


Figure 5.2: The adjusted divider is composed of two $\div 2$ circuit and one program counter

The value of K is set to 4, so we design $f_{REF} = 40$ MHz in order to satisfy the optimal

equation $f_{REF}K = R = 160$ MHz. Accordingly, the PLL output frequency is generated at intervals of 160 MHz, allowing for a total of 10 frequency outputs ranging from 800 MHz to 2240 MHz. To meet such frequency planning, the divider is designed with divisors of 20, 24, 28, 32, 36, 40, 44, 48, 52, and 56, which means that the program counter provides a continuous range of modulus from 5 to 14.

5.3 Phase Noise Consideration

In this work, the performance of phase noise is relatively worse. As mentioned in 3.4.2, the phase noise of the free-run VCO is about -91.8 dBc/Hz at 1 MHz offset. And as shown in Figure 5.3, we can see that the VCO noise actually dominates the phase noise performance of the whole PLL.

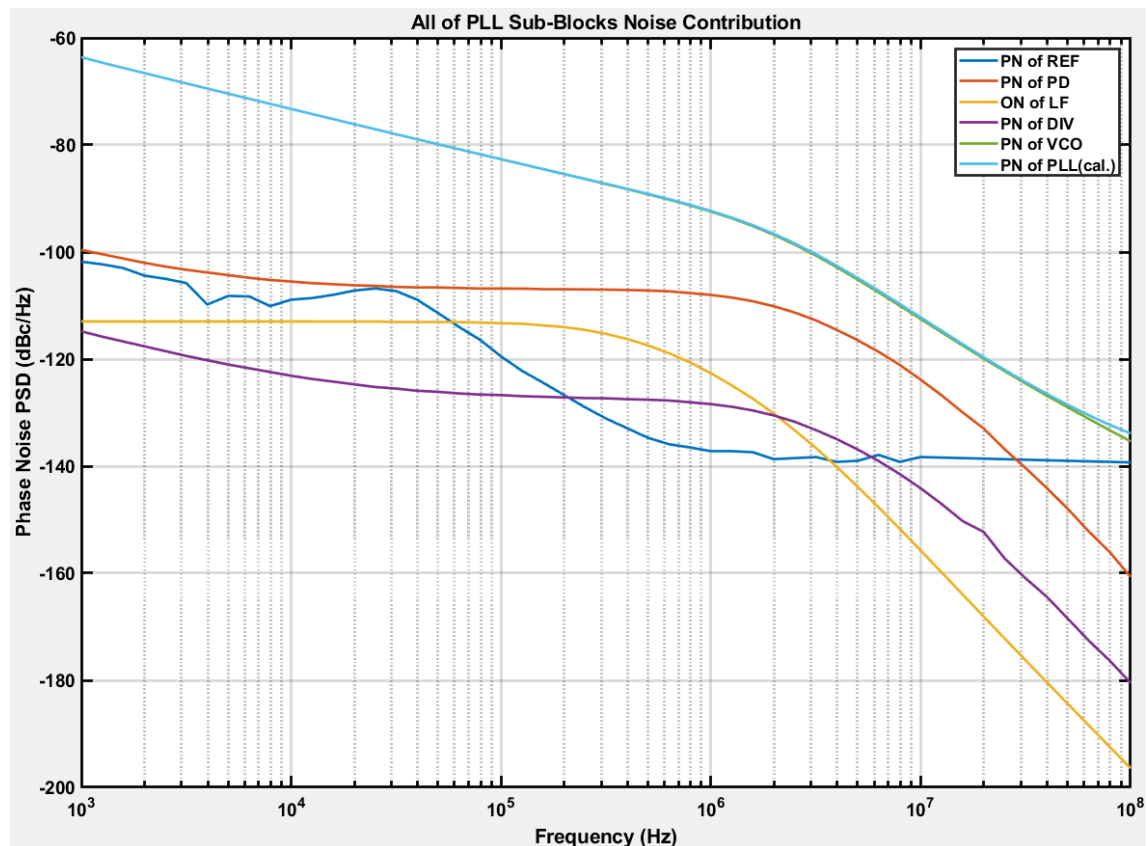
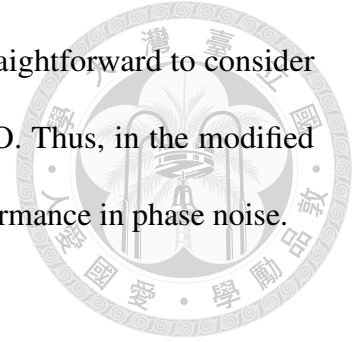


Figure 5.3: All of PLL sub-blocks noise contribution

When it comes to suppressing VCO noise in a PLL, it's straightforward to consider that a larger bandwidth can suppress more noise from the VCO. Thus, in the modified design, we will enlarge the loop bandwidth to improve the performance in phase noise.



5.4 Simulation Result

The layout of the improved PLL is shown in Figure 5.4. The area of the BSL has been reduced by nearly 35% after the optimization. The decoder and the adder for the lower bound have been eliminated. The area of the PLL divider has also been reduced, resulting in an overall reduction in the PLL area of approximately 30%.

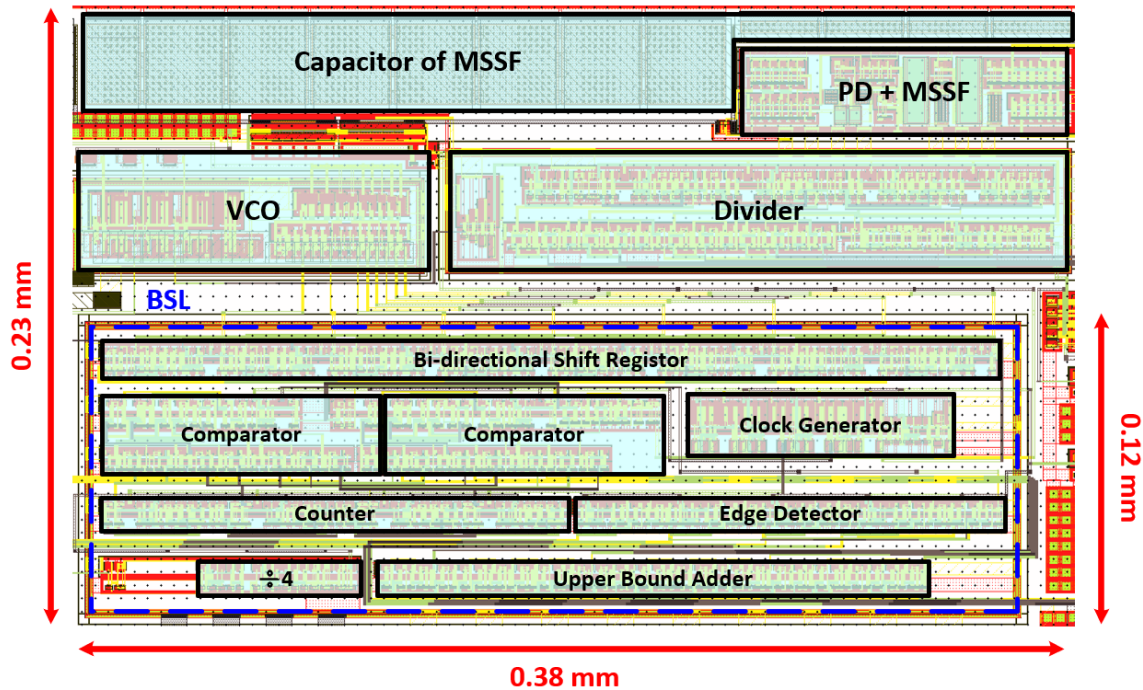


Figure 5.4: Layout of the improved PLL

The transient simulation result is shown in Figure 5.5. The PLL is set to operate at the lowest frequency of 840 MHz at first. At $t = t_1$, the divisor is switched to 56, triggering the activation of the BSL. During this process, the VCO is disconnected from the main loop,

and V_{ctrl} is set to a constant value. At $t = t_2$, the output band falls within the acquisition range of the target frequency, prompting the BSL to deactivate and reconnect the VCO to the main loop. Eventually, V_{ctrl} settles to a proper value that locks the PLL to its maximum output frequency of 2240 MHz.

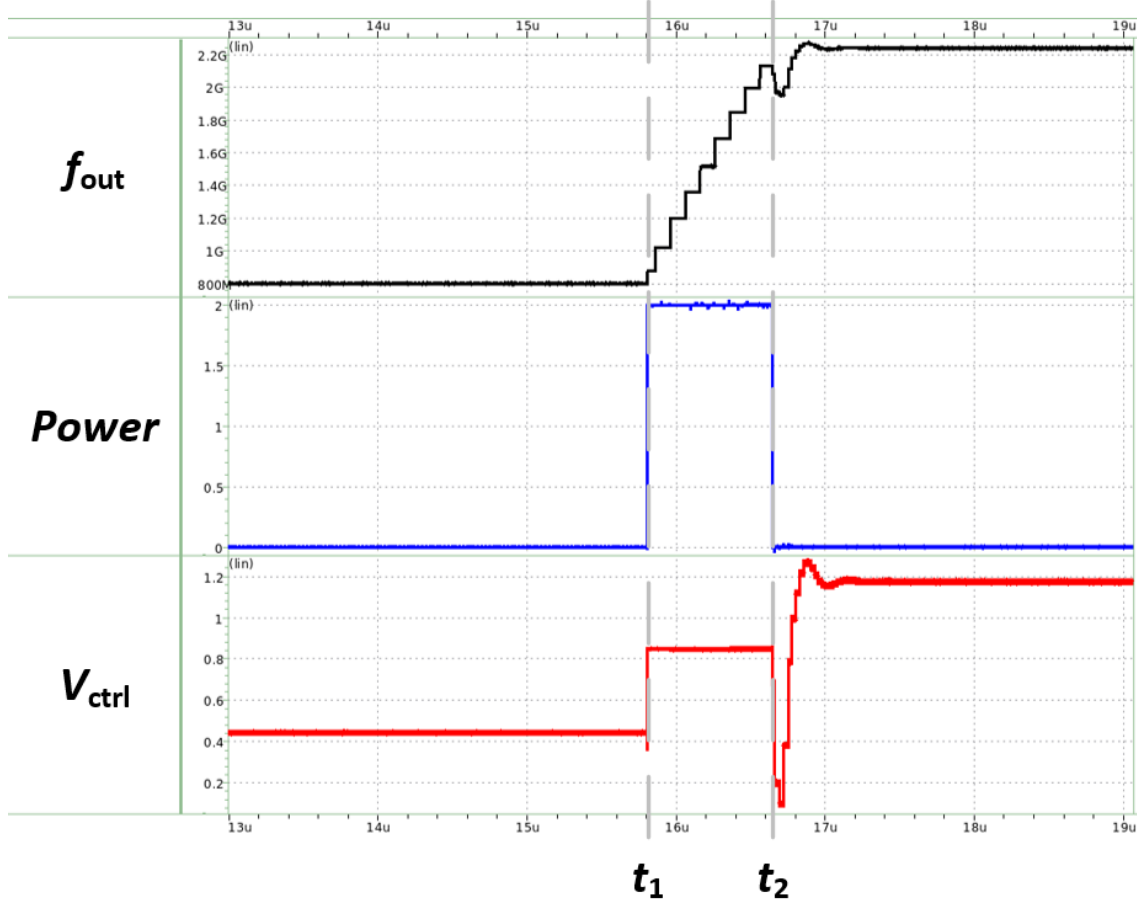


Figure 5.5: Transient simulation result of the improved system

Recall the equation (3.12), it defines the time required by the band-selecting process in the worst-case is $T_{REF} + A \times T_{REF} + T_{REF}$. While we have 8 bands in total and $T_{REF} = 25$ nsec for $f_{REF} = 40$ MHz, the time it takes in the worst-case would be 850 nsec. In the transient simulation result, it takes a total of 843 nsec to switch from the lowest frequency band (band0) to the highest frequency band (band8), which matches our previous calculation. Note that compared to the original chip, the f_{ref} has increased

fourfold from 10 MHz to 40 MHz, so the time required for the band-selecting process has decreased by fourfold. The time needed to lock after switching divisors has been reduced.

The output frequency design ranges from 800MHz to 2240MHz, with a frequency output every 160MHz, totaling 10 frequencies. The spectrum is shown in Figure 5.6, demonstrating the spectrum of the highest frequency of 2240 MHz.

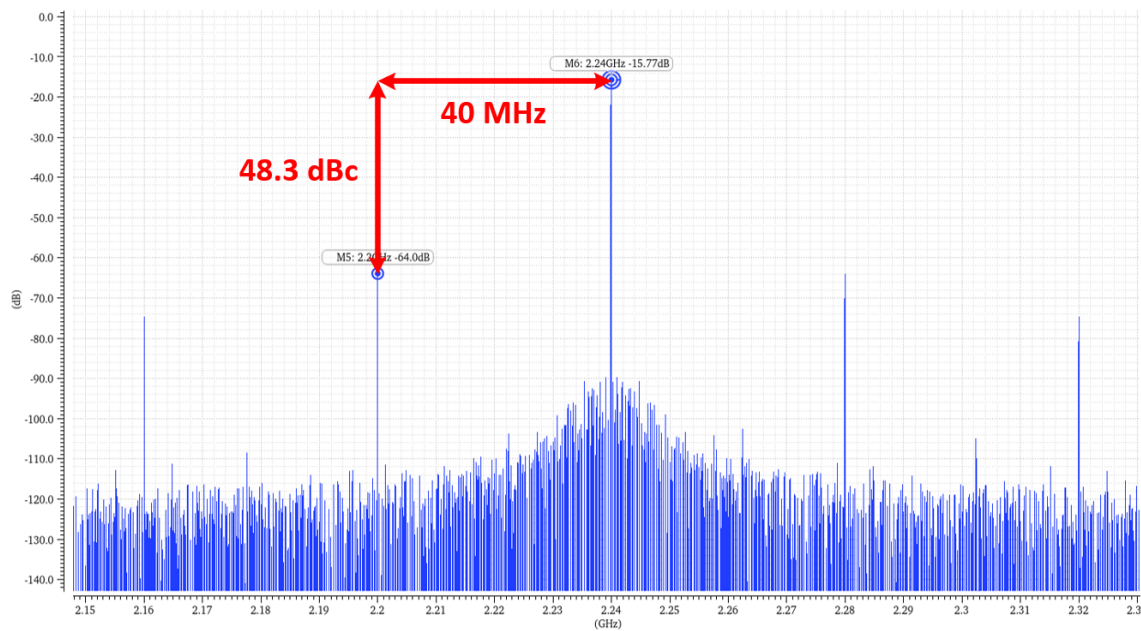


Figure 5.6: Output spectrum when $f_{out} = 2240$ MHz

We designed a wider bandwidth to further suppress the contribution of VCO noise to the overall PLL phase noise. Figure 5.7 and Figure 5.8 show the phase noise simulation result for two output frequencies. When $f_{out} = 800$ MHz, phase noise is -99.5 dBc/Hz at 1 MHz frequency offset. When $f_{out} = 2240$ MHz, phase noise is -96.8 dBc/Hz at 1 MHz frequency offset. With the same VCO as the previous work, the performance of phase noise improves significantly because of the redesign of bandwidth.

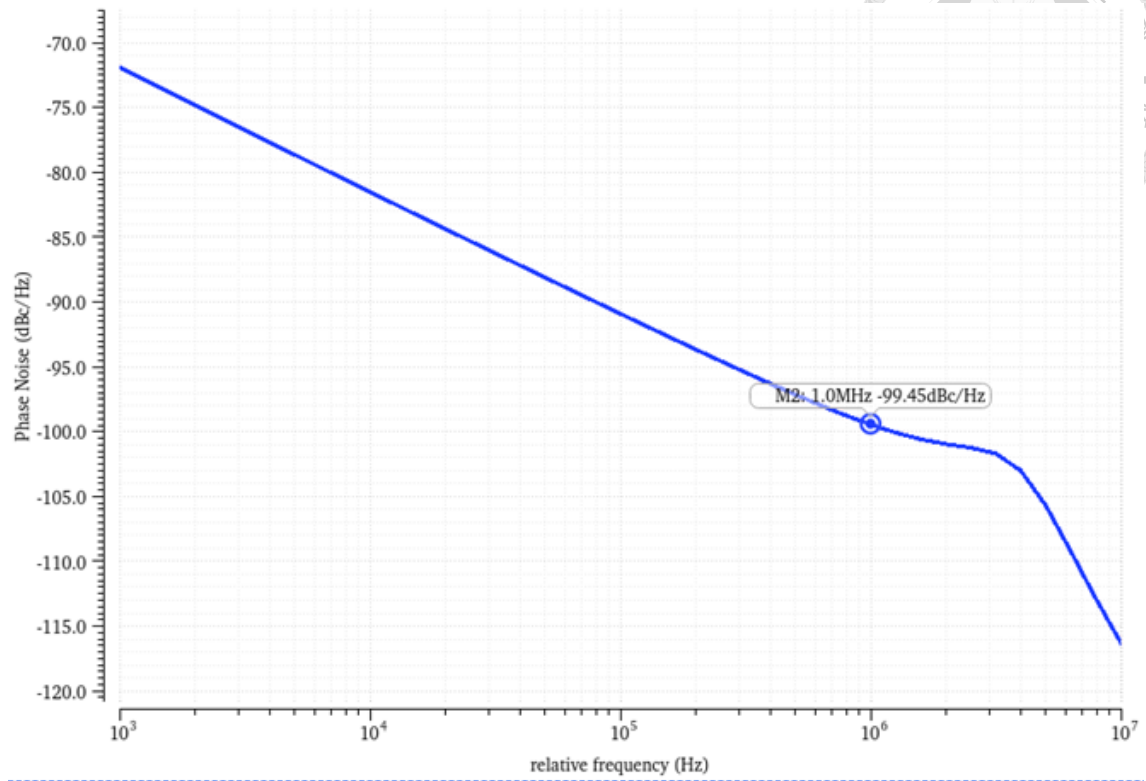
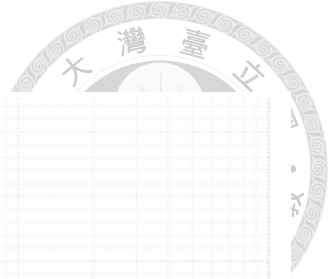


Figure 5.7: Phase noise when $f_{out} = 800$ MHz

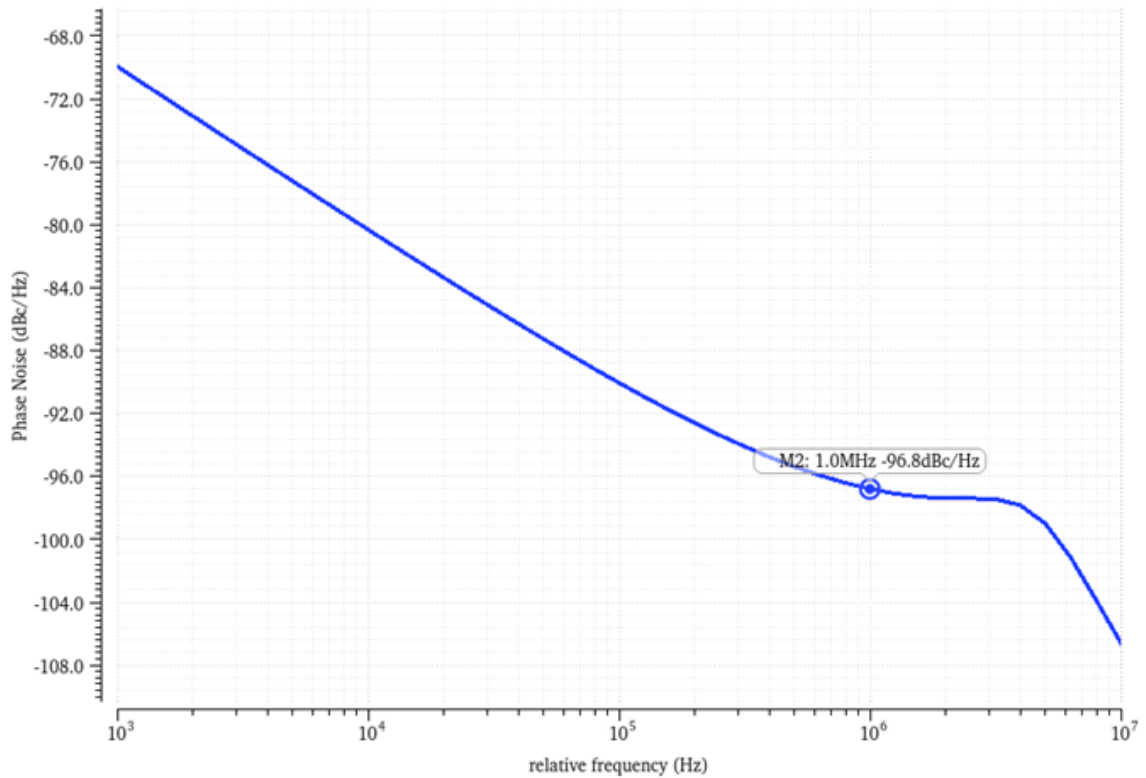
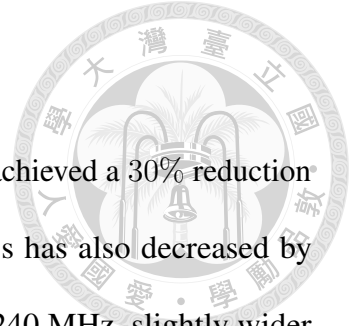


Figure 5.8: Phase noise when $f_{out} = 2240$ MHz

5.5 Summary

After optimizing the parameters and frequency planning, we achieved a 30% reduction in the core area. The time taken for the band-selecting process has also decreased by fourfold. The output frequency now spans from 800 MHz to 2240 MHz, slightly wider than the previous PLL design. Besides, we redesigned the loop bandwidth to suppress the noise from VCO further. The performance of phase noise improved for about 5 dB after modification. The layout of this chip has been completed, and it was taped out in February of this year. Measurements will be conducted after the chip fabrication is completed in June of this year.





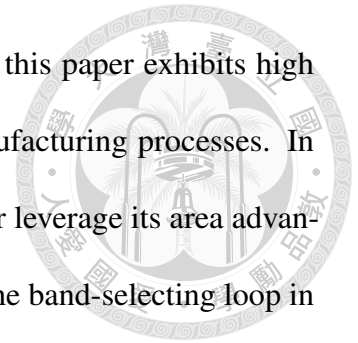
Chapter 6

Conclusion

This thesis presents a novel band-selecting loop to achieve a wide tuning range with multiple modulus type-I PLL. In Chapter 2, the principle and characteristics of the type-I PLL are introduced. Although this architecture is relatively compact and inherently stable, its limited acquisition range restricts its practical applications. Additionally, there is a trade-off between the acquisition range and the reference spur. In Chapter 3, we propose the band-selecting mechanism to overcome the trade-off. It calibrates the frequency to within the acquisition range before the PLL initiates locking. This mechanism enables the realization of a wide tuning range type-I PLL. On top of that, we cover each block in the PLL in detail. Subsequently, a chip fabricated using TSMC 180-nm CMOS technology is implemented to validate the feasibility. The experimental results, including the transient response, reference spur, and phase noise, are demonstrated in Chapter 4. Finally, in Chapter 5, we optimized the system's parameter selection and frequency planning, resulting in a PLL design that can further reduce the area by 30%.

The band-selecting mechanism proposed in this thesis can be applied not only to type-I PLLs but also to other systems requiring frequency calibration, such as injection-locked

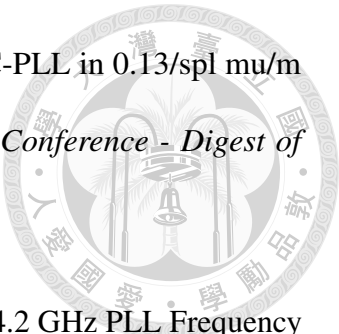
oscillators. Furthermore, the band-selecting loop presented in this paper exhibits high programmability and is highly adaptable across different manufacturing processes. In advanced manufacturing processes, this architecture can further leverage its area advantages. In the future, it is also worth considering implementing the band-selecting loop in a cell-based design flow manner to accelerate the design process. Moreover, automated synthesis tools can typically generate layouts with even smaller areas.





Reference

- [1] L. Kong and B. Razavi, “A 2.4 GHz 4 mW Integer-N Inductorless RF Synthesizer,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 3, pp. 626–635, 2016.
- [2] B. Razavi, *Design of analog CMOS integrated circuits*. Tsinghua University Press Co., Ltd., 2005.
- [3] A. Grebene and H. Camenzind, “Phase locking as a new approach for tuned integrated circuits,” in *1969 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, vol. XII, 1969, pp. 100–101.
- [4] A. Sharkia, S. Mirabbasi, and S. Shekhar, “A Type-I Sub-Sampling PLL With a $100 \times 100 \mu\text{m}^2$ Footprint and 255-dB FOM,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 12, pp. 3553–3564, 2018.
- [5] H. Rategh, H. Samavati, and T. Lee, “A CMOS frequency synthesizer with an injection-locked frequency divider for a 5-GHz wireless LAN receiver,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 5, pp. 780–787, 2000.
- [6] P. Hanumolu, M. Brownlee, K. Mayaram, and U.-K. Moon, “Analysis of charge-pump phase-locked loops,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 9, pp. 1665–1674, 2004.

- 
- [7] R. Gu, A.-L. Yee, Y. Xie, and W. Lee, “A 6.25GHz 1V LC-PLL in 0.13/spl mu/m CMOS,” in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, 2006, pp. 2442–2451.
- [8] T. Wu, P. K. Hanumolu, K. Mayaram, and U.-K. Moon, “A 4.2 GHz PLL Frequency Synthesizer with an Adaptively Tuned Coarse Loop,” in *2007 IEEE Custom Integrated Circuits Conference*, 2007, pp. 547–550.
- [9] S. P. Bruss and R. R. Spencer, “A 5GHz CMOS PLL with low KVCO and extended fine-tuning range,” in *2008 IEEE Radio Frequency Integrated Circuits Symposium*, 2008, pp. 669–672.
- [10] B. Çatlı, A. Nazemi, T. Ali, S. Fallahi, Y. Liu, J. Kim, M. Abdul-Latif, M. R. Ahmadi, H. Maarefi, A. Momtaz, and N. Kocaman, “A Sub-200 fs RMS jitter capacitor multiplier loop filter-based PLL in 28 nm CMOS for high-speed serial communication applications,” in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, 2013, pp. 1–4.
- [11] F. Gardner, “Charge-Pump Phase-Lock Loops,” *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1849–1858, 1980.
- [12] F. Song, Y. Zhao, B. Wu, L. Tang, L. Lin, and B. Razavi, “16.5 A Fractional-N Synthesizer with 110fsrms Jitter and a Reference Quadrupler for Wideband 802.11ax,” in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2019, pp. 264–266.
- [13] S.-Y. Cho, S. Kim, M.-S. Choo, J. Lee, H.-G. Ko, S. Jang, S.-H. Chu, W. Bae, Y. Kim, and D.-K. Jeong, “A 5-GHz subharmonically injection-locked all-digital PLL with

complementary switched injection,” in *ESSCIRC Conference 2015 - 41st European Solid-State Circuits Conference (ESSCIRC)*, 2015, pp. 384–387.

[14] B. Razavi, *Design of CMOS phase-locked loops: from circuit level to architecture level*. Cambridge University Press, 2020.

[15] S. Shekhar, *Wideband frequency synthesizers*. University of Washington, 2008.

[16] B. Razavi, “Principles of data conversion system design,” (*No Title*), 1994.

[17] Y. Sun, J. Li, Z. Zhang, M. Wang, N. Xu, H. Lv, W. Rhee, Y. Li, and Z. Wang, “A 2.74–5.37GHz boosted-gain type-I PLL with ± 152012 *IEEE Radio Frequency Integrated Circuits Symposium*, 2012, pp. 181–184.

[18] A. Sharkia, S. Aniruddhan, S. Mirabbasi, and S. Shekhar, “A Compact, Voltage-Mode Type-I PLL With Gain-Boosted Saturated PFD and Synchronous Peak Tracking Loop Filter,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 43–53, 2019.

[19] R. K. Nandwana, T. Anand, S. Saxena, S.-J. Kim, M. Talegaonkar, A. Elkholy, W.-S. Choi, A. Elshazly, and P. K. Hanumolu, “A Calibration-Free Fractional-N Ring PLL Using Hybrid Phase/Current-Mode Phase Interpolation Method,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 882–895, 2015.

[20] A. Sai, T. Yamaji, and T. Itakura, “A 570fsrms integrated-jitter ring-VCO-based 1.21GHz PLL with hybrid loop,” *2011 IEEE International Solid-State Circuits Conference*, pp. 98–100, 2011.

