國 立 交 通 大 學

電子工程學系電子研究所

碩 士 論 文

應用於 Serial ATA 之全數位展頻時脈產生器
及數位可程式化之高斯時脈產生器

All Digital Spread Spectrum Clock Generator
for Serial ATA Application
& Digital Programmable Gaussian Clock Generator

研 究 生：莊　　立

指導教授：周 世 傑

中　華　民　國　九十九　年　三　月

應用於 Serial ATA 之全數位展頻時脈產生器
及數位可程式化之高斯時脈產生器

All Digital Spread Spectrum Clock Generator
for Serial ATA Application
& Digital Programmable Gaussian Clock Generator

研 究 生：莊　立　　Student：Li Chuang

指導教授：周世傑　　Advisor：Prof. Shyh-Jye Jou

國　立　交　通　大　學

電子工程學系　電子研究所碩士班

碩　士　論　文

A Thesis

Submitted to Department of Electronics Engineering &

Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

In

Electronics Engineering

March 2010

Hsinchu, Taiwan, Republic of China

中華民國　九十九　年　三　月

應用於 Serial ATA 之全數位展頻時脈產生器及數位可程式化之高斯時脈產生器

研究生：莊立　　指導教授：周世傑 博士

國立交通大學

電子工程學系 電子研究所碩士班

摘要

在這篇論文中,我們聚焦於利用輸入調變方式的全數位時脈展頻電路.為了產生較快頻率以及較小頻率展頻量的調變時脈,我們提出了一個新的 Domino 調變方式.我們產生出來的調變時脈.可以提升至100MHz以及5000ppm的展頻量.相比之下,之前發表的論文其調變時脈只有 23MHz 以及 3%.在架構設計上,我們也提出一個新的粗細混何的數位延遲線.此架構相可以節省 330%以及 383%的功率消耗及面積.我們提出的數位展頻時脈產生器(AD-SSCG)有成為矽智財(IP)的潛力.因為其電路都是由數位電路所設計的.這個 AD-SSCG 矽智財被用來與一個 1.2GHz 以及一個 3GHz 的 PLL 在做模擬,都可以成功產生展頻.最後我們把 AD-SSCG 與一個 3GHz 的 PLL 利用 UMC 90 奈米 1P9M 的製程實現.AD-SSCG 所佔的面積是 335um × 105um 其功率消耗是 2.9mW.利用 hspice post-sim 模擬 3GHz 的 EMI 下降量為 22dB.

最後,我們提出的一個數位高斯時脈產生電路,它是設計給 CDR 作測試使用.高斯時脈生器使用了高斯雜訊產生器,並把這個雜訊轉換成一個高斯時脈.產生

的高斯時脈有著調整它 jitter 大小的能力,以用來驗證 CDR 在不同環境的需要. 我們將提出的高斯時脈產生器合成於 FPGA 發展板.我們可以將資料驗証量提升到 10^{12} 筆,以驗證 CDR 的錯誤率.

All Digital Spread Spectrum Clock Generator for Serial ATA Application & Digital Programmable Gaussian Clock Generator

Student：Li Chuang Advisor：Prof. Shyh-Jye Jou

Department of Electronics Engineering
Institute of Electronics
National Chiao Tung University

ABSTRACT

In the thesis, we focus on the AD-SSCG (All Digital Spread Spectrum Clock Generator) modulation method with input reference. In order to achieve higher frequency and less frequency deviation, we propose a new Domino modulation method. We can improve the modulated clock to 100MHz and 5000ppm of frequency deviation as compared to 23MHz modulated clock and 3% of frequency deviation is published before. In the architecture design, we propose a novel Coarse-Fine DDLi (Digital Delay Line) structure, it improve the power and area by 330% and 383% than traditional structure. The AD-SSCG has the potential to become an IP because all of the circuits are deigned by digital circuit. This IP is used to work with a 1.2GHz and a 3GHz PLL, and both of them can spread spectrum successfully. Finally, our AD-SSCG and a 3GHz PLL are implemented with UMC-90-CMOS 1P9M process.

The area and power of AD-SSCG are respectively 335um × 105um and 2.9mW, and the EMI reduction of 3GHz PLL is 22dB by hspice post-sim simulation.

Finally, a proposed digital programmable Gaussian clock generator is designed for CDR (Clock and Data Recovery Circuit) testing. The Gaussian clock generator uses Gaussian noise generator to transforms a clock to a Gaussian clock. The generated Gaussian clock has the ability of controlled jitter to verify CDR with different environments. By using the proposed Gaussian clock generator on a FPGA board, we can easily verify the performance of the CDR to 10^{12} data to check the Bit ERROR Rate (BER).

致謝

謝謝這幾年周世傑老師對我的指導以及照顧，老師是一個對自己要求嚴格的，總是讓我可以學到許多態度，希望老師未來可以激發出更多交大人的潛能，並且能夠健康快樂的工作、生活。除此之外，也非常感謝李鎮宜教授、蘇朝琴教授以及黃錫瑜教授能夠來參加我的畢業口試，給予我論文更多的幫助以及指導。

要感謝的人非常的多，首先感謝的是 TOTORO 學長，學長對我在作研究上很大方的幫助，除此之外也很感謝明賢學長、誠文學長、庭楨學長、紹維學長、spice 學長、阿樸學長，也很感謝跟我同屆的同學舒榮、ABC、祥蛙、運祥，也很開心因為認識范姜、阿九以及奕瑋，使我在實驗室又有更多的樂趣。最後也很感謝其他實驗室的朋友勛哲、篤雄、致煌、至中、魏胖、阿光，有了大家研究所的生活可是又開心又充實。

當然我要謝謝父母奶奶弟弟，家人對我的支持，在這一路上不斷的栽培及鼓勵我，使我一路可以專心讀書，也才有機會考上交通大學，更有機會得到這個交大碩士的學位，之後出社會，會把從交大這邊學到的態度，繼續往前走。

莊立
新竹交大
2010.3.10

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

Technology scaling has dramatically increased the amount of computation. The increased computation of digital circuit in SOC will require higher bandwidth to transmit signal. As a result, the design of chip I/O has became increasingly sophisticated, with multi-Gb/s bandwidth to provide higher computation mount of systems and networks. The higher speed serial link is composed of (1) transmitter, (2) channel, and (3) receiver. The transmitter use high speed PLL with multi-phase to transmit high bandwidth signal. When the signal passes through the channel which often has poor frequency response at high frequency, the signal will decrease at the part of high frequency. In the receiver end, we could use equalizer to compensate the loss due to the channel. The equalized signal will pass through CDR (clock data recovery circuit) to recover correct data which is the original bitstream from transmitter end. The consideration of system design includes the noise of transmitter signal, channel design, package design, signaling method, equalization, and CDR specification. Another important issue shall be highlighted. When PLL generates higher frequency clock, it often brings more EMI (electron magnetic interference) problem. The solution to implement of high speed PLL, we can spread this high speed clock with specified frequency deviation. The power will be scattered to wider spread

bandwidth to reduce EMI problem. The specification of SATA [1] also sets the limitation that the EMI reduction needs to be larger than 7dB. However, the new challenge will be generated both in transmitter and receiver at the same time because the CDR must recover the data with intrinsic deterministic jitter due to that the transmitter use spread clock to transmit the signal.

1.2 Motivations and Goals

There is a publication [12] of SSCG with input reference modulation on JSSC 2007. The paper names "All Digital Spread Spectrum Clock Generator" because all of the circuits are designed by digital circuits. This modulation method has significant difference among SSCG modulation methods. All of the other modulations do the modulation profile in PLL loop with analog approach. But paper [12] only uses digital approach outside of the PLL loop to generate spread spectrum clock. After reading this paper [12], we think this kind of modulation method can be implemented as an IP (Intellectual Property). Thus, firstly we want to implement our AD-SSCG (all digital spread spectrum clock generator) which can have different spread modes, like up and down, and have different frequency deviation for different required specifications. Secondly, we think the frequency of the generated modulated_CLK of paper [12] is too slow, our goal is to increase the frequency of the modulated_CLK. Thirdly, for SATA application, the frequency deviation is less than paper [12], so we need to present a solution to achieve less frequency deviation.

Besides the research of AD-SSCG, we also interest in the topics of CDR testing. Often, the BER (Bit Error Rate) will be estimated by two methods (1) after chip tape-out, we use equipment BERT to generate high speed clock with Gaussian noise (2) before chip tape-out, we can use MATLAB to generate Gaussian clock to estimate

the BER of CDR. The method (2) will verify the system robustly but the verification platform is implemented on computer. It is limited by hardware capacity and limited memory so the testing data amount only can achieve around 10^6. In order to solve the problem of testing in pre-tape out procedure, we use Gaussian noise generator and some digital circuits to construct a Gaussian noise clock generator with the controllability of different jitter. All of these circuits are designed by digital circuits so it can be implemented on FPGA developed board. We use this new testing platform to breakthrough the amount of testing data.

1.3 Thesis Organization

Chapter 2 will introduce the basic concept of SSCG and four main kinds of modulation methods. Chapter 3 will analyze the SSCG with input reference method and make comparisons between paper [12] and ours. We will construct simulink model to compare the performance. Chapter 4 will introduce the implementation method of our AD-SSCG and design consideration. Then, the experimental results and SSCG performance comparisons are also in this chapter. Chapter 5 will show the design of programmable Gaussian clock generator. Chapter 6 is the conclusion.

Chapter 2

Basic Concept of Spread Spectrum Clock

2.1 The Background of EMI Problem

Today, electrical devices have more Electromagnetic Interference (EMI) at operational frequency. EMI will pollute radio spectrum which is caused by the radiation of unwanted radio frequency signals. EMI will occur in those electronic systems that change voltages and current rapidly. Thus, the Federal Communications Commission (FSS) in the United States has regulation rules about the maximum power of EMI [1] as shown in Fig. 2.1. The regulation rules focus on the peak power, not on the average power.

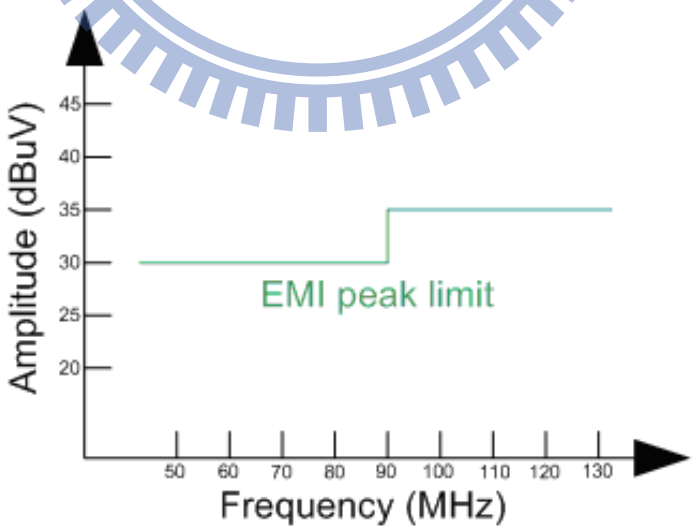


Fig. 2.1 FFC EMI peak limit [1]

In the chip design, the bandwidth of data transmission will increase continuously.

The required high speed clock will have high frequency component. The metal line in chip will suffer from the EMI phenomenon. In the field of serial link, the specification of SATA-2 [2] also defines the EMI requirement. Table 2.1 shows the SATA-2 specification. The EMI has to be reduced by 7dB at least. Then, the SATA-2 specification also defines the spreading mode, amount and modulation frequency.

Table 2.1 SATA-2 specifications [2]

Parameter	**Value**
EMI reduction	>7 dB
Spread spectrum mode	Down Mode
Spread amount	5000 ppm
Modulation frequency	30 ~ 33 KHz

2.2 Concept of Spread Spectrum Clock

The basic idea of the spread spectrum clock is to slightly modulate the frequency of clock signals and the energy of the signals will be scattered to a controllable wide range. The energy is distributed by modulating the signal slowly between two frequency boundaries. The peak energy of every harmonic component in the spectrum is decreasing after spreading spectrum. Fig.2.2 shows the signal with and without spread spectrum [3].

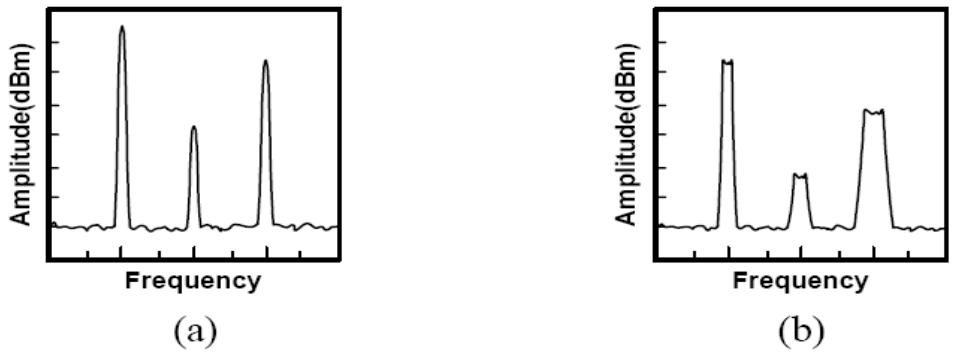


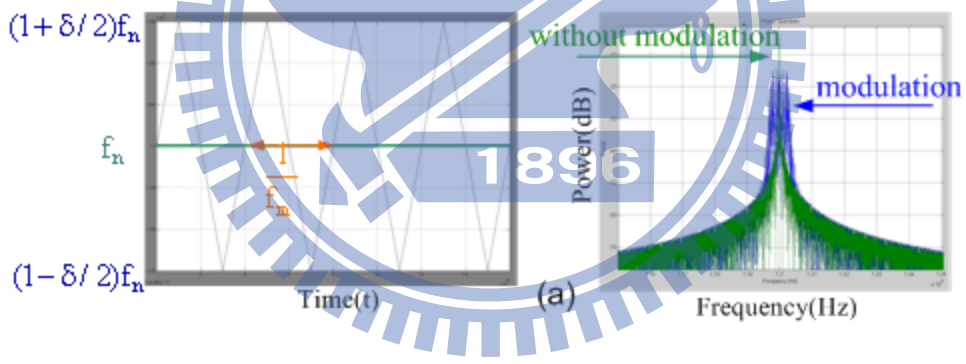
Fig. 2.2 Spectrum of (a) Un-spread spectrum signals (b) Spread spectrum signals

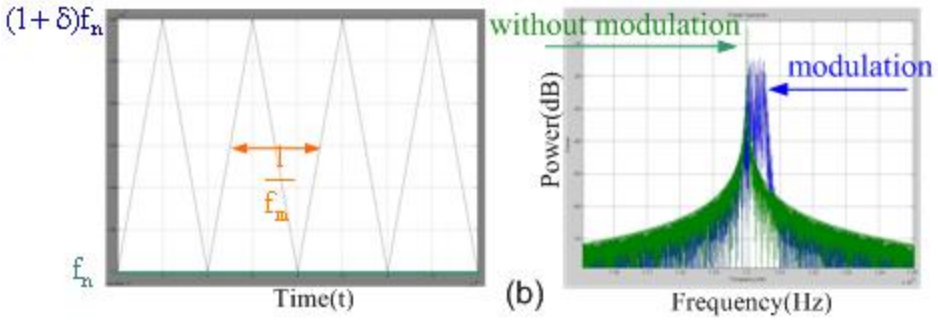
2.3 Spread Spectrum Parameters

2.3.1 Spread Spectrum Modes and Amounts

There are three main kinds of spread spectrum modes, center-spread, down-spread, and up-spread.

We define the parameter firstly. The parameter f_n is the central frequency. The parameter δ is the total amount of spreading as a relative percentage of f_n. For central-spread modulation, the frequency starts at f_n, and the frequency varies from $f_n(1+\frac{1}{2}\delta)$ to $f_n(1-\frac{1}{2}\delta)$ as shown in Fig. 2.3 (a). For the up-spread modulation, the frequency varies from f_n to $f_n(1+\delta)$ as shown in Fig. 2.3 (b). For the down-spread modulation, the frequency varies from f_n to $f_n(1-\delta)$ as shown in Fig. 2.3 (c). In Fig. 2.3, we will find parameter f_m which is defined as modulation frequency. The inverse of f_m is one period of modulation.





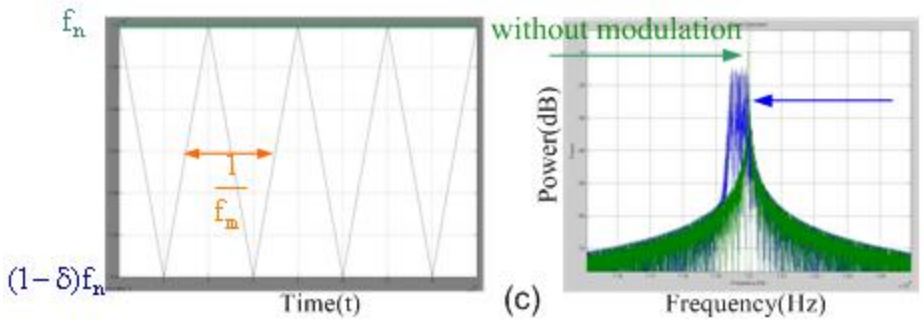


Fig. 2.3 (a) Central-spread (b) Up-spread (c) Down-spread

For SATA-2 example, δ=5000ppm with down-spread modulation, and f_m=30~33KHz. Thus, if we use f_n with 1.2GHz, the frequency of modulated clock varies from 1.2GHz to 1.194GHz. The modulation period is the inverse of 30KHz to 33KHz, that is, 33.33us to 30.30us.

2.3.2 Modulation Profile

The modulation profile is one of the important parameters that affect the performance of spread spectrum. The shape of modulation profile determines the distribution of energy. In other words, the modulation profile in time domain determines the shape of power energy in frequency domain.

Fig. 2.4 shows two common modulation profiles. For the sinusoidal modulation, we can find out that the peak in two sides of spectrum is larger than central spectrum. For the triangular modulation, the peak in two sides is larger than central spectrum, but the difference of spectrum height between two sides and central point is less than that of sinusoidal modulation. We can observe that in the sinusoidal modulation, the location time of more maximum and minimum frequency is longer than triangular modulation. In EMI standard, we care about the peak of spectrum, not average power. So, we prefer to use triangular modulation to get more EMI reduction.

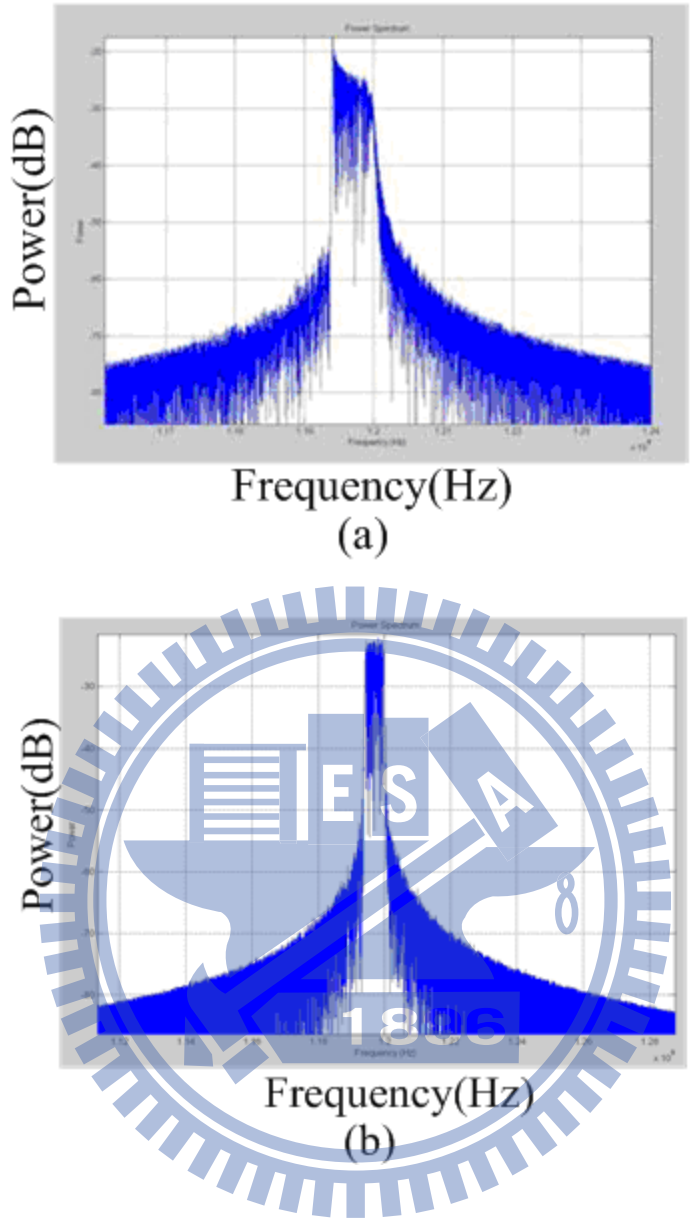


Fig. 2.4 Spectrum of (a) Sinusoidal profile (b) Triangular profile

2.3.3 Modulation Frequency

The modulation frequency is an important factor that will affect the EMI reduction performance. In the main stream, the modulation frequency is between 30 to 50KHz. The higher modulation frequency will have better EMI reduction performance as shown in Fig. 2.5 for Matlab simulation. Although the higher modulation frequency will have more EMI reduction, the modulation frequency is limited due to the timing jitter of the clock source and the tracking capability of the timing recovery in the receiver.

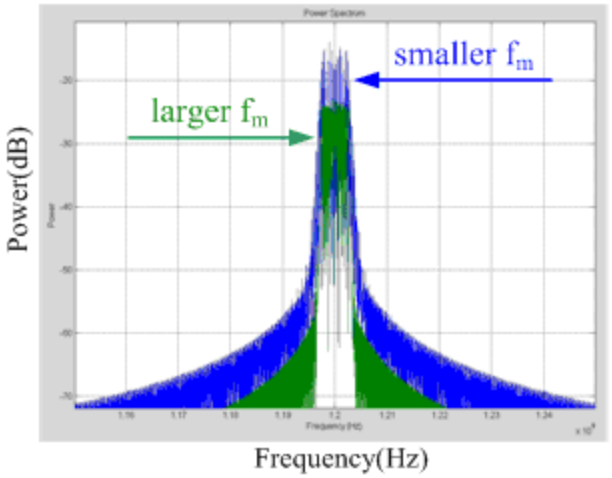


Fig. 2.5 Spread spectrum with different modulation frequency

2.3.4 Timing Impacts

In a SSCG system, the frequency varies with time as shown in Fig. 2.6 (b). For serial link application, we will use spread spectrum clock in transmitter and receiver. We must know the tolerance of the system for changing of clock. So, we need to know some important parameters, like cycle-to-cycle jitter and long term jitter [3].

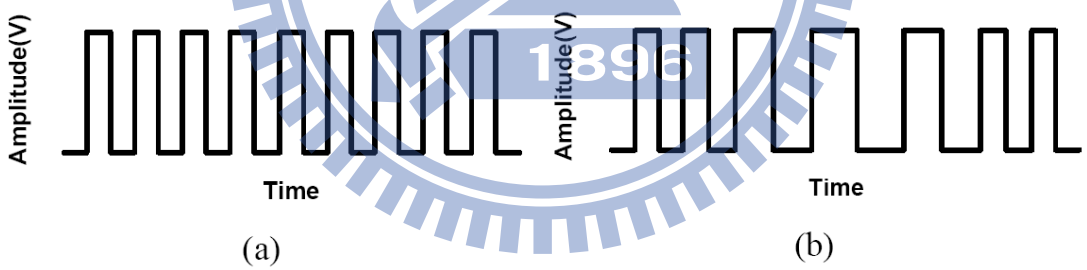


Fig. 2.6 Time domain behavior
(a)Un-spread spectrum signals (b)Spread spectrum signals

A. Cycle-to-Cycle Jitter

When we use spread spectrum clock, the frequency will vary with time. The period of every clock will increase or decrease continuously. So, every clock has different period. The definition of cycle-to-cycle jitter is the difference between presentperiod and next period. Fig. 2.7 is an example of cycle-to-cycle jitter.

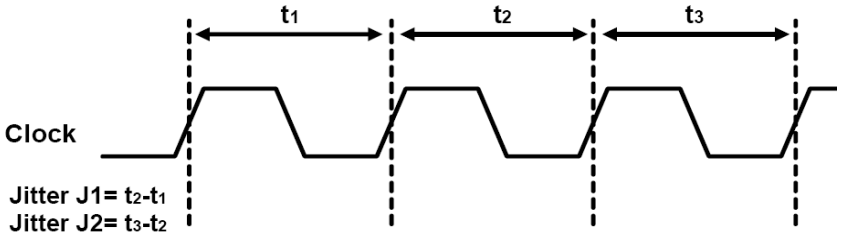


Fig. 2.7 Cycle-to-cycle jitter

We will use a formula to calculate cycle-to-cycle jitter [4] for down spread mode which is the case for SATA-2 specification. Firstly, we calculate the period difference between normal frequency and maximum modulation frequency in Equation (2.1).

$$\Delta T_{total} = \frac{1}{(1-\delta)f_n} - \frac{1}{f_n} \cong \frac{\delta}{f_n} \tag{2.1}$$

Then, we calculate the number of clocks in half time modulation. The frequency changes from f_n to $f_n(1-\delta)$ in half time modulation. Equation (2.2) will calculate the number of clocks in half time of the modulation cycle.

$$N = \frac{1}{2f_m} \Big/ \frac{1}{f_{avg}} = \frac{f_{avg}}{2f_m} \tag{2.2}$$

where f_m and f_{avg} are the modulation and average frequency of the spread spectrum clock. We adopt the triangular modulation profile to calculate the average frequency as shown in Equation (2.3).

$$f_{avg} = (1-0.5\delta) \times f_n \tag{2.3}$$

Therefore, the cycle-to-cycle jitter can be expressed as shown in Equation (2.4).

$$\begin{aligned} \Delta T_{c-c} &= \frac{\Delta T_{total}}{N} = \frac{\delta}{f_n} \times \frac{2f_m}{(1-0.5\delta)f_n} \\ &= \frac{2f_m\delta}{(1-0.5\delta)f_n^2} \end{aligned} \tag{2.4}$$

For a 1.2GHz spread spectrum clock with 5000ppm frequency deviation and 31KHz modulation frequency, the cycle-to-cycle jitter is shown in Equation (2.5).

$$\begin{aligned} \Delta T_{c-c} &= \frac{2 \times 31 \times 10^3 \times 0.5\%}{(1-0.5 \times 0.5\%)(1.2 \times 10^9)^2} = 2.15 \times 10^{-16} \\ &= 0.215 \text{ fs} \end{aligned} \tag{2.5}$$

B. Long-Term Jitter

Long-term jitter defines the maximum cycle change form its ideal position as shown in Fig. 2.8. Equation (2.6) shows the long-term jitter of clock 1.2GHz with 5000ppm frequency deviation [4].

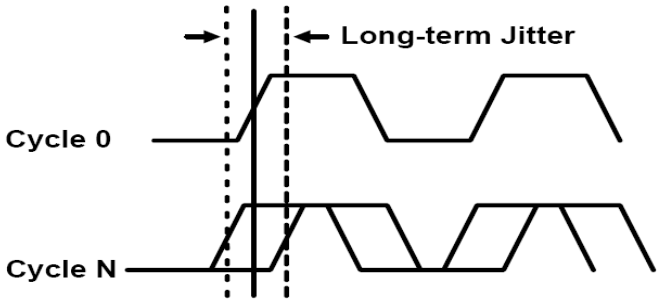


Fig. 2.8 Long-term jitter

$$\Delta T_{total} = \frac{\delta}{f_{nom}} = \frac{0.5\%}{1.2\times10^{9}} \cong 4.166\times10^{-12} \cong 4.2\ ps \tag{2.6}$$

2.4 Different Types of Spread Spectrum Clock Generator

A basic PLL is comprised of PFD, CP, LPF, VCO and Divider. Due to EMI problem, PLL need to have spread spectrum function. There are four common modulation methods [4]: (1) modulation on VCO, (2) modulation on Divider, (3) phase selection of multiphase VCO, and (4) input reference modulation.

Fig. 2.9 shows the SSCG of modulation on VCO. In this method, we often need another CP to generate voltage of triangular voltage on Vctrl as modulation profile. Because we know the relation between the Vctrl and frequency of VCO, we could know the required voltage profile to add. The paper [5-7] adopt this modulation method. However, this modulation method will be serious distorted with process variation. For example, the Kvco curve and the current amount of CP will be changed due to process variation. It will reduce the performance of EMI.

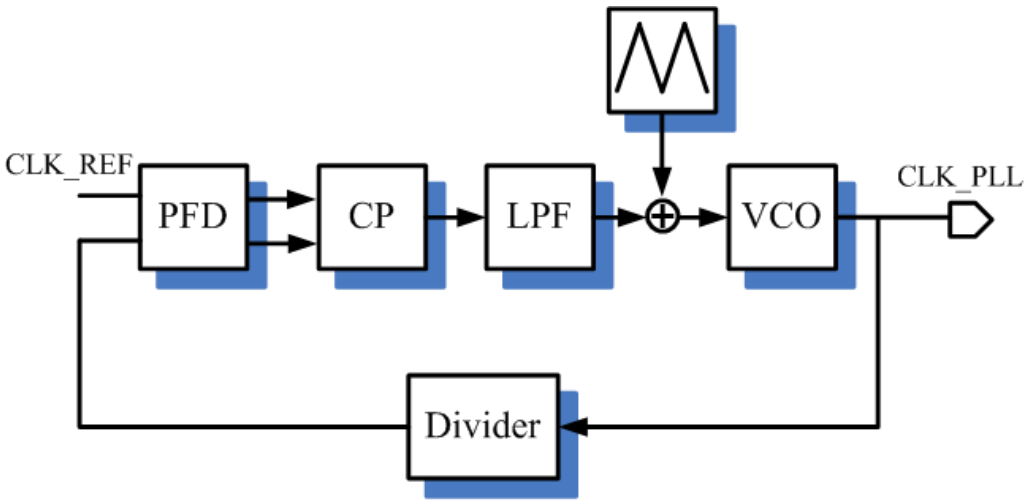


Fig. 2.9 SSCG of modulation on VCO

Fig. 2.10 shows the SSCG of modulation on Divider. The mechanism is that the Divider changes the divider ratio, and the frequency of VCO will change from original frequency to spreading spectrum frequency. Often, there are many steps between original divider ratio (N) and spreading spread divider ratio (N+n). More steps of division ratio will make the triangular profile like an ideal one and have better EMI reduction. Paper [8] and [9] use fractional delta-sigma multi-modulus divider and FDMP (Frequency Dual-Modulus Prescaler) to increase the steps. The feature of this method needs to have a Divider with large division modulus. The divider ratio has an impact on the loop characteristic of the PLL. Thus, it is difficult to have N that is shooting for both the specification of PLL and SSCG.

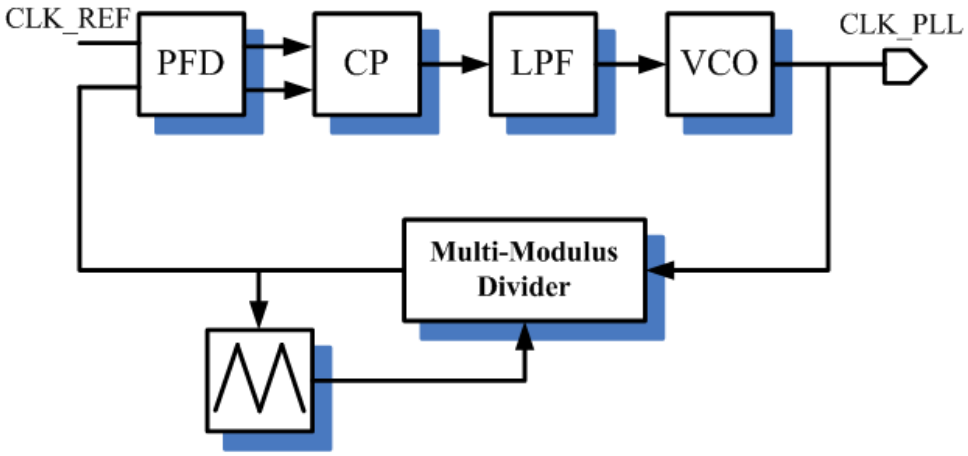


Fig. 2.10 SSCG of modulation on divider

Fig. 2.11 shows the SSCG of phase selection of multi-phase VCO. It needs a MUX to connect the multi-phase clock of VCO. We can design the numbers of cycles to jump one phase to determine frequency spread spectrum amount. The mechanism is that when MUX jump to the next phase, it will let the Divider be triggered fast. When PFD will find the feedback CLK_DIV is faster than CLK_REF, PFD will show DW signal to reduce the frequency of VCO. The gradual jumping phase will make PLL have spread frequency function. Paper [10] and [11] adopt this modulation method. The structure of this modulation method needs a multi-phase VCO. In general, if we want to achieve better EMI reduction performance, we need an interpolator to cascade to VCO to generate extra phases. It will consume more power on this structure. Besides, due to the phase jumping, we should avoid the occurrence of glitch.

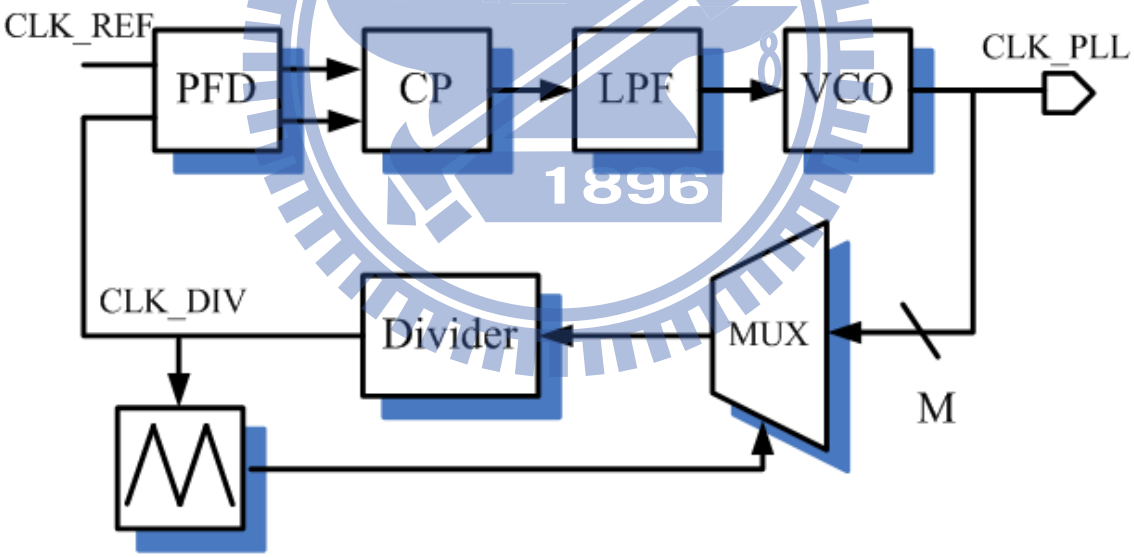


Fig. 2.11 SSCG of phase selection

Fig. 2.12 shows the SSCG with input reference modulation. Firstly, the reference clock signal (CLK_REF) passes through a SSCG block and SSCG generates modulated_CLK. A PLL is used to track this modulated_CLK and will have spread spectrum ability. The frequency deviation and modulation frequency of PLL will be

decided by the SSCG. Paper [12] shows that the PLL tracks a 23MHz modulated_CLK and finally the 148.5MHz VCO has 13dB EMI reduction with 3% frequency deviation. In this SSCG modulation method, PLL designer can focus on the design of PLL, and don't need extra effort to consider spread spectrum function. The only concern of the PLL is to have the ability to track the modulated_CLK. Moreover, due to the operation of SSCG is on F_{req_REF} which is much smaller than F_{req_PLL}, this SSCG can be designed by all digital circuit approach (AD-SSCG).

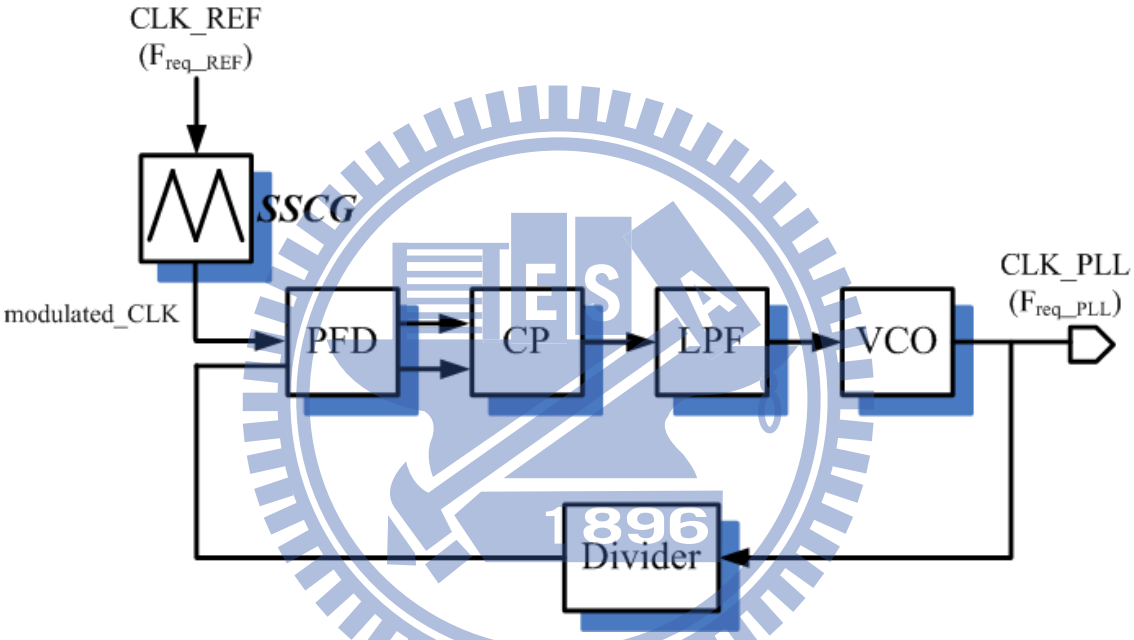


Fig. 2.12 SSCG of modulation on input reference

The thesis will focus on the method of modulation on input reference and design this SSCG by all digital circuits (AD-SSCG). Besides, we will improve the performance compared to paper [12], for example, we could generate higher frequency modulated_CLK with less frequency deviation. The AD-SSCG will become an essential IP for PLL. Firstly, the AD-SSCG is implemented by digital circuit shall have programmable capability. For example, we can design different (1)frequency deviation, (2)modulation frequency, (3)frequency of modulated_CLK, and (4)spread mode (up and down spread spectrum). Secondly, this AD-SSCG IP can

be used to any PLL only if the CLK_REF frequency of PLL is the same as modulated_CLK frequency. Thirdly, the AD-SSCG can be portable to the next generation process because the circuit is implemented by digital circuit. Finally, the design of PLL will have more challenge in advanced process. PLL designers will make more effort to ensure their PLL will work on every process corner. If PLL designers need to add spread spectrum function to the PLL, it will need more design time. It will reduce the time to market and difficulty of designing PLL. This AD-SSCG IP can solve the problem.

Our goal is to design a programmable AD-SSCG. In one of the operation modes, it is designed for SATA-2 specification. The modulated_CLK of paper [12] is 3% frequency deviation. Obviously, it isn't used for SATA-2 because the frequency deviation is not small enough. Our AD-SSCG is designed with multi-frequency deviation, that are 5000ppm, 10000ppm, and 15000ppm, and multi-spread modes, that are up and down spread modes. The down spread spectrum with 5000ppm frequency deviation mode, is designed for SATA-2.

Chapter 3
Propose All Digital
Spread Spectrum Clock Generator

3.1 Introduction

We will introduce the SSCG method of input reference modulation. In this kind of SSCG, we will design a block in front of the PLL. When the CLK_REF passes through this block, the CLK_REF will be modulated. Then the modulated_CLK will be used as CLK_REF to the input of PLL. The PLL will track modulated_CLK and have spread spectrum function. The feature of spread spectrum clock of PLL is decided by this block. Because this block is designed by all digital circuits, so we call it " AD-SSCG" (All Digital Spread Spectrum Clock Generator).

In section 3.2, we will describe and analyze the SSCG method of input reference modulation of JSSC 2007 [12]. Then, we try to use the modulation method to meet our design specification, that is, modulated_CLK is 100MHz with 5000ppm frequency deviation. However, the modulation method [12] has some timing limitations on chip implementation. So we propose a new modulation method to achieve our target. In section 3.3, we will introduce the realization method of AD-SSCG. In section 3.4, we will build up a formula for the proposed SSCG modulation method. This formula can help us design for different requirements. In the

end of this section, we show an example of our design. In section 3.5, we make summary of our specification. In section 3.6, we will build the AD-SSCG behavior model in Simulink platform and show the spread spectrum performance of the proposed AD-SSCG. In section 3.7, we will use MATLAB simulation to decide the optimized parameters of proposed modulation to achieve optimized EMI reduction. Finally, in section 3.8, we compare the EMI reduction and modulation profile of paper [12] and ours.

3.2 The Concept of Domino AD-SSCG

According to paper [12], the method to generate modulated_CLK is that every N clock has the same frequency. There are 2M groups which have N clocks, and the number of steps in frequency deviation is M which doesn't include the original frequency, that is f_n. The modulated frequency is from f_n, $f_n+\Delta f$, $f_n+2\Delta f$, …, to $f_n+M\Delta f$, then to $f_n+(M-1)\Delta f$,…., to $f_n+2\Delta f$, $f_n+\Delta f$. The modulation method is shown in Fig. 3.1.

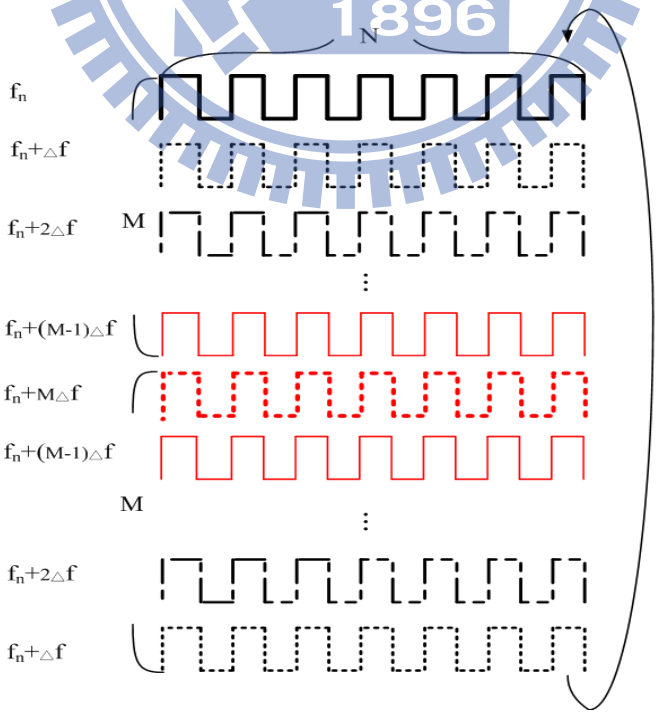


Fig. 3.1 The modulation method of paper [12]

In [12] the reference clock is 27MHz. The period of reference clock is around 37037ps. The period of spread spectrum clock with down spread and 3% frequency deviation is 38182ps. The delay time of each delay BUF is 200ps. It uses phase switching to generate different frequency clock. If the modulated frequency is f_n+ 1×Δf, the amount of phase switching is 1, and so on. Based on the modulation method of paper [12], the number of steps in frequency deviation is 6 as shown in Equation (3.1).

$$\text{Period}_{27\text{MHz}(1\text{-}3\%)}\text{-Period}_{27\text{MHz}}=38182\text{ps}-37037\text{ps}=1145\text{ps}$$

$$\text{Number of Frequency}=\frac{1145\text{ps}}{\text{BUF}}=\frac{1145\text{ps}}{200\text{ps}}\cong 6 \tag{3.1}$$

SSCG block can generate 7 modulation frequency including reference clock frequency, that are, 27MHz, 27×(0.995) MHz, 27×(0.990) MHz, 27×(0.985)MHz, 27×(0.980)MHz, 27×(0.975)MHz, and 27× (0.970) MHz.

Firstly, we adopt this modulation method to generate higher frequency clock which is 100MHz with 3% frequency deviation and have the same number of frequency steps. From Equation (3.2), we need a 50ps delay BUF.

$$\text{Period}_{100\text{MHz}(1\text{-}3\%)}\text{-Period}_{100\text{MHz}}=10309\text{ps}-10000\text{ps}=309\text{ps}$$

$$\text{BUF}=\frac{309\text{ps}}{\text{Number of Frequency}}=\frac{309\text{ps}}{6}\cong 50\text{ps} \tag{3.2}$$

Secondly, we use this modulation method to generate 100MHz modulated_CLK with 5000ppm frequency deviation to meet SATA-2 specification. The requirement of the frequency deviation is severer because 5000ppm (0.5%) is smaller than 3%. If the numbers of frequency steps is also 6. From Equation (3.3), we need a 8.3ps delay BUF.

$$\text{Period}_{100\text{MHz}(1\text{-}0.5\%)}\text{-Period}_{100\text{MHz}}=10050\text{ps}-10000\text{ps}=50\text{ps}$$

$$\text{BUF}=\frac{50\text{ps}}{\text{Frequency Step}}=\frac{50\text{ps}}{6}\cong 8.3\text{ps} \tag{3.3}$$

Table 3.1 summaries the parameters of the modulated_CLK with this modulation method. We can find three things. Firstly, if we want to make higher frequency modulated_CLK, the required delay BUF is smaller. Secondly, if we want to have modulated CLK with less frequency deviation, the required delay BUF is also smaller. Thirdly, if we want to more frequency steps, the required delay of BUF is smaller too. These three conditions will set the delay BUF too difficult to design. For example, in UMC 90nm CMOS process, the minimum delay time of BUF is around 30ps. If we want to use this method to generate modulated_CLK which is 100MHz with 5000ppm frequency deviation, we need a 8.3ps delay BUF. We have to use advanced process to generate delay BUF with smaller delay, like 65nm or 40nm process, because the RC constant in advanced process is smaller. We think this modulation method is only suitable for low frequency modulated_CLK and large frequency deviation. In order to meet the goal of 100MHz with 5000ppm frequency deviation in 90nm CMOS process, we have to propose a new modulation method.

Table 3.1 BUF delay in different clock frequency and frequency deviation

modulated_CLK frequency	Frequency deviation	Steps of frequency	BUF
23MHz	3%	6	200.0ps
100MHz	3%	6	50.0ps
100MHz	0.5%	6	8.3ps

Our proposed new modulation method is shown in Fig. 3.2. We use two different frequency clocks to achieve modulated_CLK. One is the original clock which is CLK_REF (f_n), and the other one is modulated clock with maximum frequency ($f_n + \delta f_n$). Here, we call every N clock in Fig. 3.2 as a group and there are 2M groups in a modulation cycle. For the ease of developing the behavior, we make M=N in the following sections. In the 1st group, all of the clocks are original clocks. In the

2nd group, only one clock is replaced by the modulated clock. In the 3rd group, two clocks are replaced by the modulated clock. The number of modulated clocks gradually increases one-by-one in every succeeding group. In the N-th group, N-1 original clocks are replaced by the modulated clocks. This is the first half modulation cycle. We call this "Domino modulation method" since it like the Domino that after N cycles, one more clock becomes modulated clock. This is the first half of the modulation cycle. In the N+1-th group, all of the original clocks are replaced by the modulated clocks. From N+2-th to 2N-th groups, the number of modulated clocks in a group gradually decreases one-by-one. Finally, in 2N-th group, there is only one modulated clock in this group. This is the post half of the modulation cycle. Then it will continue to execute the first half of the modulation cycle, and the 1st to the 2N–th groups form one modulation cycle.

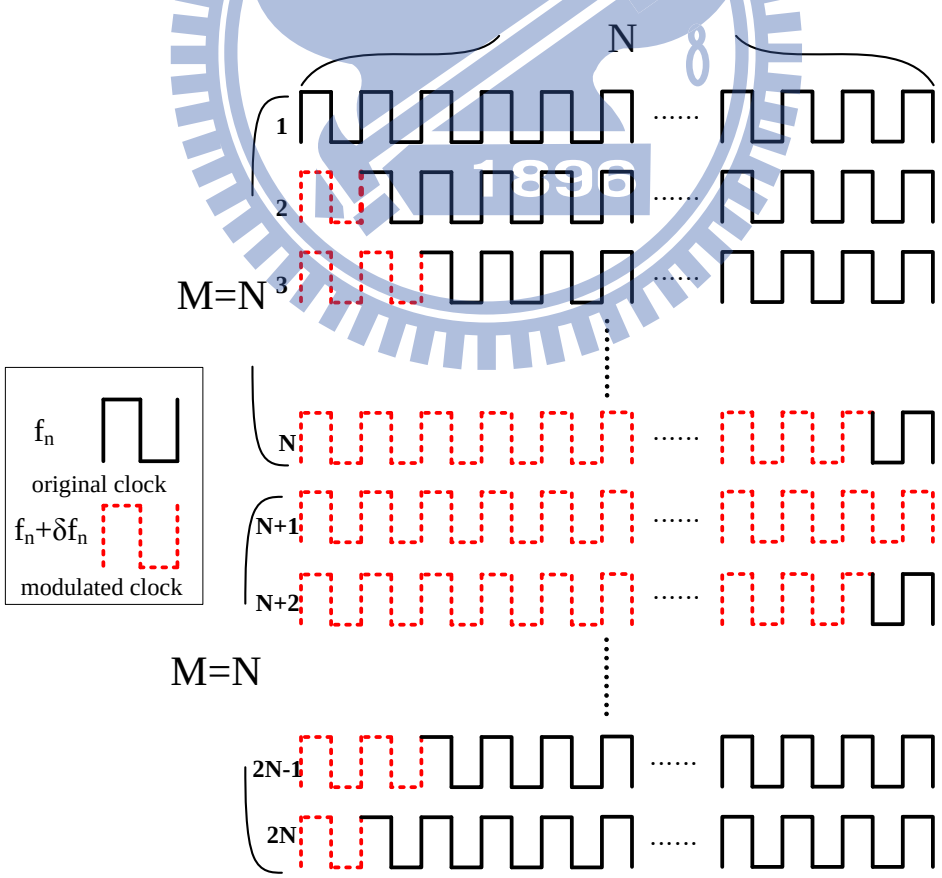


Fig. 3.2 The Proposed Domino modulation method

We can observe this modulation scheme from another point of view. We calculate the average frequency of each group. The first group is f_n. The second group is $f_n + \frac{1}{N} \times \delta f_n$. The third group is $f_n + \frac{2}{N} \times \delta f_n$. The average frequency of modulated_CLK in the 1st to the N+1-th groups is from f_n to $f_n + \delta f_n$. Then, the average frequency in the N+2-th to the 2N-th group is from $f_n + \frac{N-1}{N} \times \delta f_n$ to $f_n + \frac{1}{N} \times \delta f_n$. The total frequency deviation is δf_n. The average frequency of modulated_CLK is from f_n to $f_n + \delta f_n$, then to f_n to form a cycle, and the increasing or decreasing step of average frequency in each group is $\frac{1}{N} \times \delta f_n$. This modulation profile is a triangular function.

When the δf_n is positive, the modulated_CLK will perform up spread spectrum function. Then, when the δf_n is negative, the modulated_CLK will perform down spread spectrum function.

The concept of the proposed Domino modulation scheme is a little like the SSCG of phase selection of multi-phase VCO (in chapter 2, Fig. 2.11). The phase selection modulation uses phase selection to subtract a phase from the VCO clock and the generated CLK_DIV is a modulated_CLK. The PFD will find out the difference between CLK_DIV and CLK_REF. Then, Charge Pump will generate extra current to change the voltage control node of VCO, then frequency of VCO will change. On the contrary, we use CLK_REF to subtract or add a phase to generate modulated_CLK which is the output of AD-SSCG. and the frequency of CLK_DIV is fixed. Then, the PFD will find out the difference between modulated_CLK and CLK_DIV, and finally VCO will also achieve spread spectrum function. When the AD-SSCG block turn on the spread spectrum function to generate modulated_CLK, PLL will track the modulated_CLK so that PLL will have the same feature of spread spectrum like

modulated_CLK. For example, when the division ratio is 12, and modulated modulated_CLK is 100MHz with 0.5% frequency deviation and 30KHz modulation frequency. The PLL will have 0.5% frequency deviation with 30KHz modulation frequency but the frequency is 1.2GHz.

Finally, the proposed Domino modulation scheme relaxes the requirement of delay BUF as compared with [12]. For example, if we design a 100MHz modulated_CLK with 0.5% frequency deviation. The required delay BUF based on paper [12] modulation is 8.3ps. However, the required delay BUF based on our Domino modulation method is just 50ps

3.3 The Design of the Domino AD-SSCG

3.3.1 The Design of Modulated Clock

In this section, we will generate the modulated clock that we need in Domino modulation method. In all digital circuit approach, the simple method to adjust the width of clock is phase switching. So we need to design a multi-phase delay line. We connect delay BUF to perform a DDLi (Digital Delay Line) as shows in Fig. 3.3. The delay time of every delay BUF is T_{BUF}.

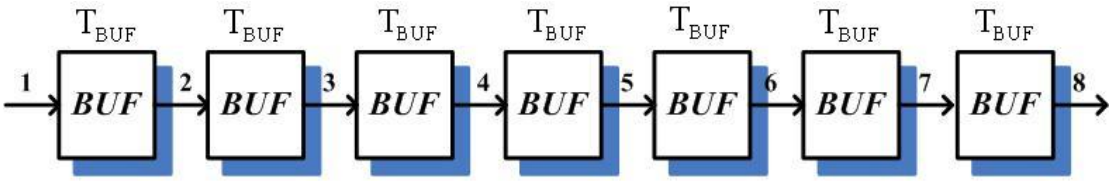


Fig. 3.3 Digital Delay Line (DDLi)

In order to have up and down spread spectrum function, we need two kinds of modulated clocks. In the following content, we discuss how to complete these two modulation clocks which have wider and narrower period.

Case 1: The modulated clock with wider period

When a square clock waveform passes through the DDLi, we know every node

has different phase. Every node has a T_{BUF} timing difference as shown in Fig. 3.3. We connect 8 nodes to a 8-1 MUX, and we assume the delay time of the 8-1 MUX is zero. In case 1, the phase selection starts at phase 1, and the direction of phase switching is from phase 1 to phase 8. The timing of phase switching is that when we find out a rising or a falling edge at present node, we will switch to next phase. As shown in Fig. 3.4, the thick line is our phase selection path. After 7 switching times, we can get modulated clock with wider period. The wider modulated clock is the summation of thick line of as shown in Fig. 3.4.

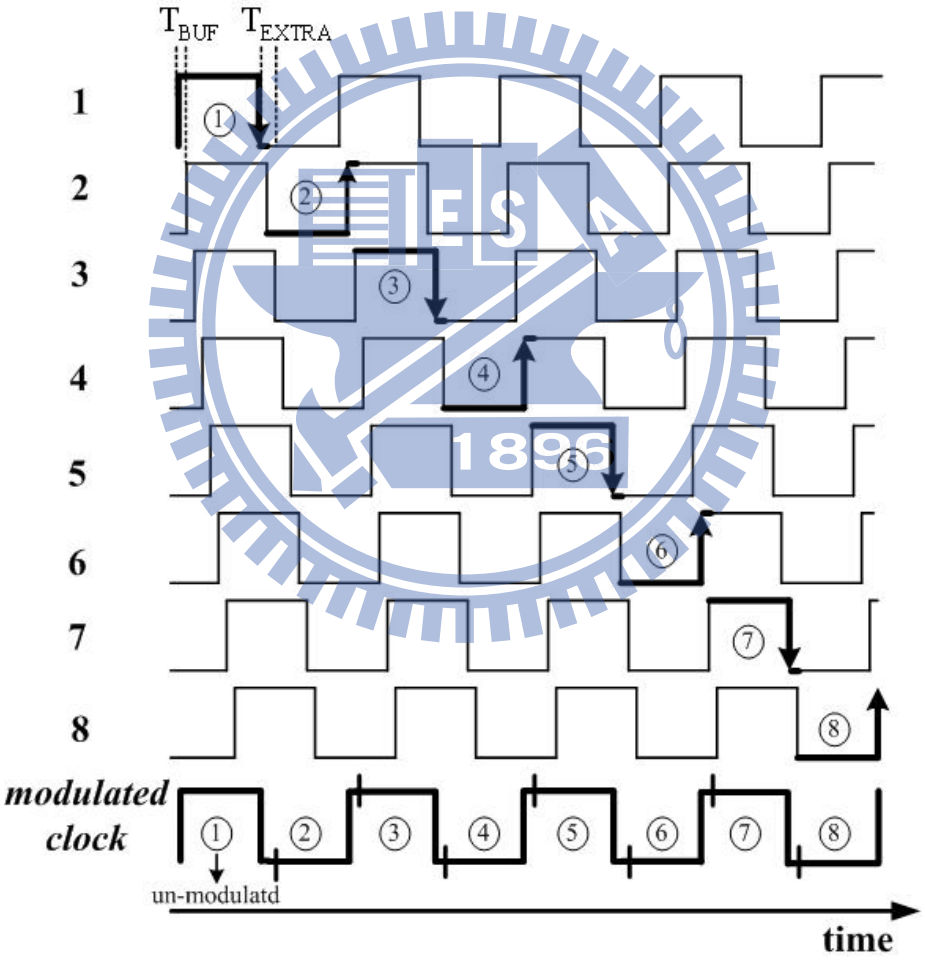


Fig. 3.4 Modulated clock with wider period

There is a design consideration. Because the phase switching is from a "leading phase" to a "lagging phase", the instant phase switching after finding out a rising or a falling edge will generate glitch. As shown in Fig. 3.4 when we find out a rising or

falling edge, we need to delay T_{EXTRA} to switch to next phase to avoid glitch. The glitch will not happen when T_{EXTRA} is larger than T_{BUF}.

Case 2: The modulation clock with narrower period

In case 2, the phase selection starts at the phase 8. And the direction of phase switching is from phase 8 to phase 1. In the same way, when we find out a rising or a falling edge, we switch to next phase. There is no glitch in this case because the phase switching is from "lagging phase" to "leading phase". As shown in Fig. 3.5, the coarse line is our phase selection path. After 7 switching times, we can get modulated clock with narrower period. The modulated clock is the summation of thick line as shown in Fig. 3.5.

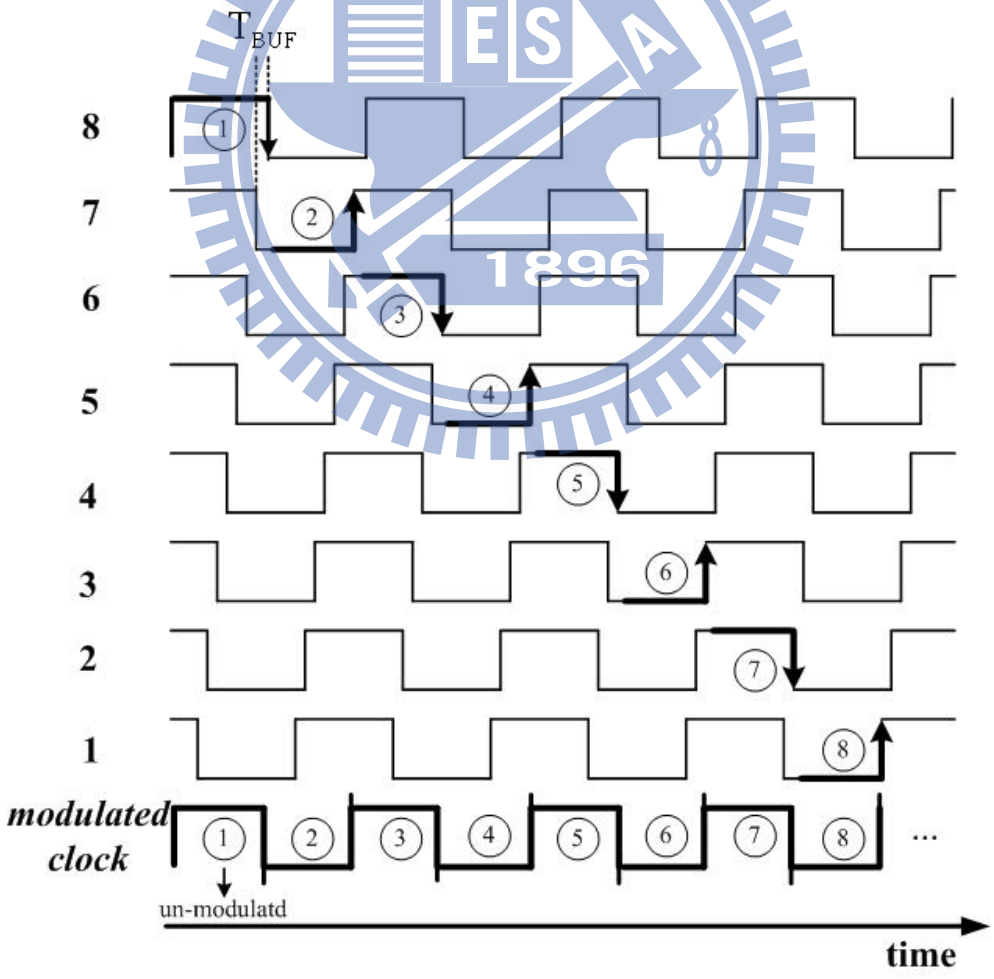


Fig. 3.5 Modulated clock with narrower period

The modulated clock can be generated by phase switching. When we design case 1, we still need to avoid occurring glitch. However, we need a lot of modulated clock in the procedure of one cycle of Domino modulation method. It means we need infinite phase to let us have next phase to switch continuously. In order to generate infinite phase, we connect infinite BUFs to complete a DDLi. This implementation method of DDLi is not practicable on chip, so we need solve this problem. We adopt the idea from paper [12], and discuss this structure in next section.

3.3.2 The Solution to Overcome Infinite Digital Delay Line

A DDLi with infinite delay is not a practical way to implement on chip. Based on paper [12], there is a method to achieve the required DDLi. We also use this method to implement our AD-SSCG. Firstly, we connect N BUFs for first part of DDLi and the delay length is T_{CLK_REF}. T_{CLK_REF} is a period of CLK_REF, that is, the input clock of AD-SSCG. Then, the second part of the DDLi is just several BUFs. We connect these two parts to form complete DDLi. Thus, the delay length of the DDLi is little more than one period (T_{CLK_REF}). The delay of each BUF is T_{CLK_REF}/N . Fig. 3.6 shows the DDLi we describe. Here, we use 4 extra BUFs for the second part of DDLi as an example to explain the behavior.

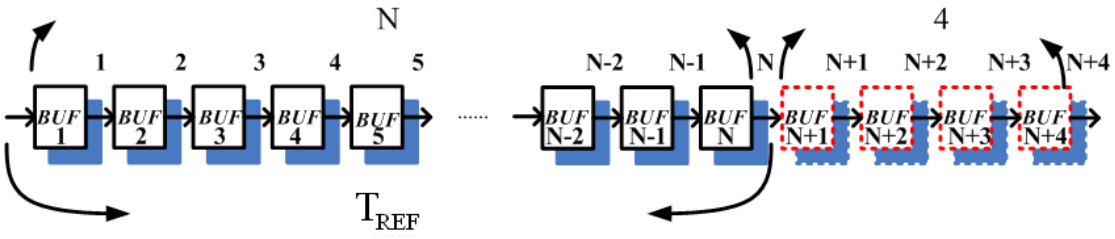


Fig. 3.6 The DDLi with delay time which is little more than one period

When we operate AD-SSCG in down spread spectrum mode, we need to generate

wider modulated clock. In this mode, the direction of phase switching is from left to right. We assume that we start at phase 1. After many times of phase switching, when we switch to phase N+4, that is, the last phase of DDLi. We have no phase to switch. However, the designed DDLi is little more than one period of CLK_REF, there will be two clock rising edges on the DDLi at the same time as shown in Fig. 3.7 (a). The "next clock" will appear at phase 4 and the distance between "present clock" and "next clock" is T_{REF}. The "present clock" can be replaced by the "next clock". We can treat "next clock" as "present clock" with different position. Right now, the switching phase is changed from N+4 to 4, and we have next phase to switch. The position of phase selection is from the end of the DDLi to the head of the DDLi.

In the same concept, when we operate AD-SSCG in up spread spectrum mode, we need to generate narrower modulated clock. In this mode, the direction of phase switching is from right to left. We assume that we start at phase N+4 and the phase 1 is the last phase to switch. After N+3 times of phase switching, the phase 1 is selected, and there is no phase to switch. However, we will find that another clock rising edge appears on the DDLi at phase N+1 as shown in Fig. 3.7 (b). The distance between "present clock" and "previous clock" is also T_{CLK_REF}. We replace "previous clock" to "present clock". Right now, the switching phase is changed from phase 1 to phase N+1, and we have next phase to switch. The position of phase selection is from the head of the DDLi to the end of the DDLi.

We summarize the concept of the procedure. In normal condition, that is we have phase to switch, we just switch to next phase. However, in no switching phase condition, we will find another clock rising edge on special designed DDLi. We can detect the position of another clock rising edge, and switch to the position of new clock rising edge. Finally, we use the new clock to continue to do phase switching.

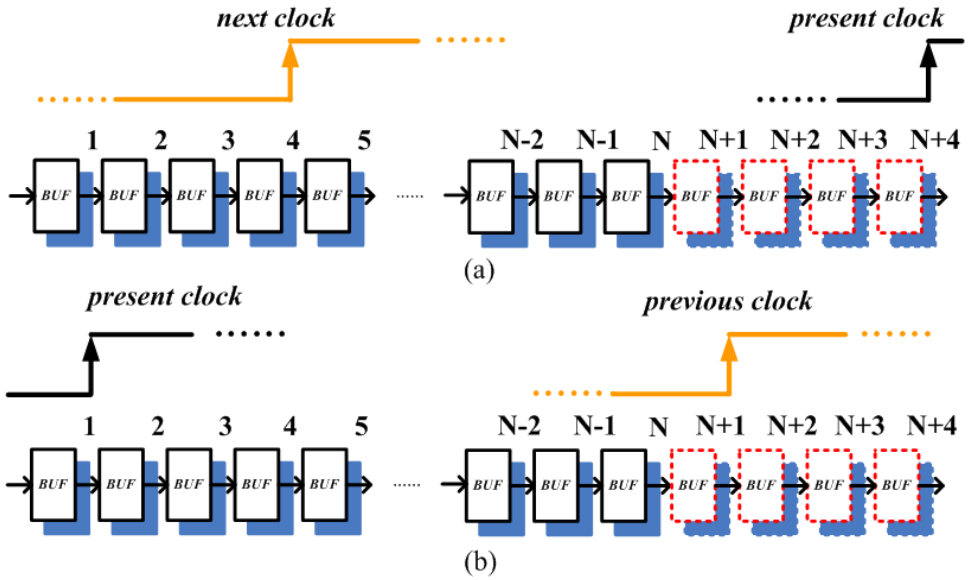


Fig. 3.7 (a) "next clock" and "present clock" on DDLi
(b) "present clock" and "previous clock" on DDLi

In order to find another clock rising edge, we need a circuit to do the action. We can use DFFs to detect the rising edge. As shown in Fig. 3.8, DFFs are connected to BUFs which locates at two sides of the delay line. The D input of DFFs is connected to the output of BUFs. We can use the output of the selected phase as the clock of DFFs. In Fig. 3.8, the selected phase is N. In the situation of no phase to switch, the new clock rising edge will appear in left or right side of DDLi. So we add extra DFFs only at two sides of DDLi.

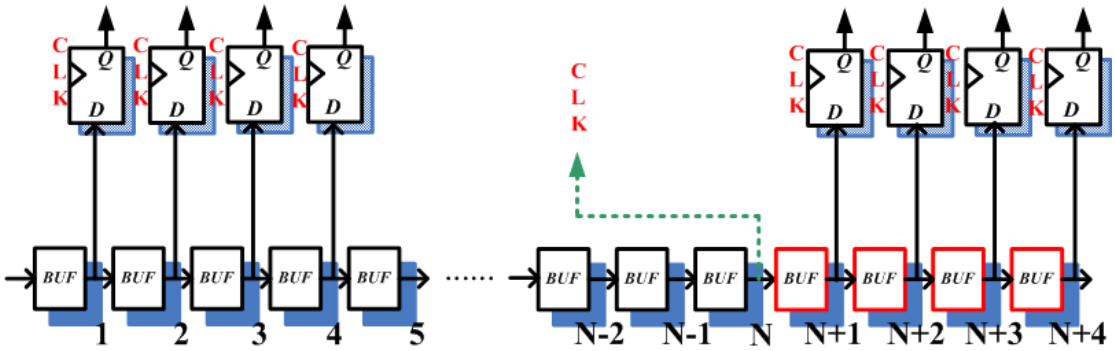


Fig. 3.8 DDLi and rising edge detection circuit

3.3.3 The Dummy Delay Line Structure

The DDLi originally has two functions. One is to be used as a multi-phase DDLi to execute phase switch, and the other one is providing edge detection circuit. Here, we separate these two functions into two parts as shown in Fig. 3.9. Because (1) if the nodes of DDLi connect too many circuits, the required delay of T_{BUF} isn't easy to complete. In practical circuit, each node of DDLi also will connect to a (N+4)-to-1 MUX. We don't want DFFs to increase loading. Because the detection function is only used when there is no phase to switch, the detection circuit of DDLi can be separated. And the CLK_REF only passes through the dummy DDLi used for detection circuit when we require. (2) It can save power consumption. Because of above two reasons, we adopt DDLi and dummy DDLi structure. The DFFs which are connected to two sides of dummy DDLi are the edge detection circuit. The left side DFFs are designed for down spread spectrum, and the right side DFFs are for up spread spectrum. The dummy DDLi is only used when there is no phase to switch. So, we will design a power saving mechanism for dummy DDLi to decide the turn on time of dummy DDLi. In normal condition, CLK_REF will not pass through dummy DDLi.

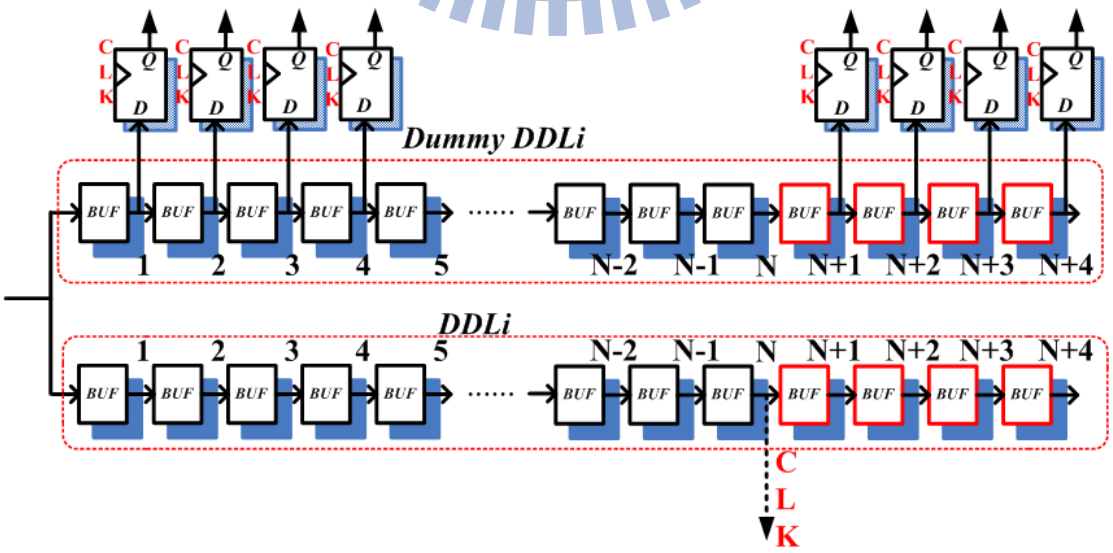


Fig. 3.9 DDLi and dummy DDLi

Fig. 3.9 has one problem due to unbalanced loading. The delay BUF of dummy DDLi should be the same as DDLi. However, the output loading of each delay BUF is not the same on the dummy DDLi due some nodes are connected to DFFs. It will cause the delay time of each BUF is not uniform. In order to solve problem of unequal output loading, Fig. 3.10 is our solution. Each delay BUF connects to two series "INV". Because each delay BUF has the same loading, the delay of each BUF is uniform.

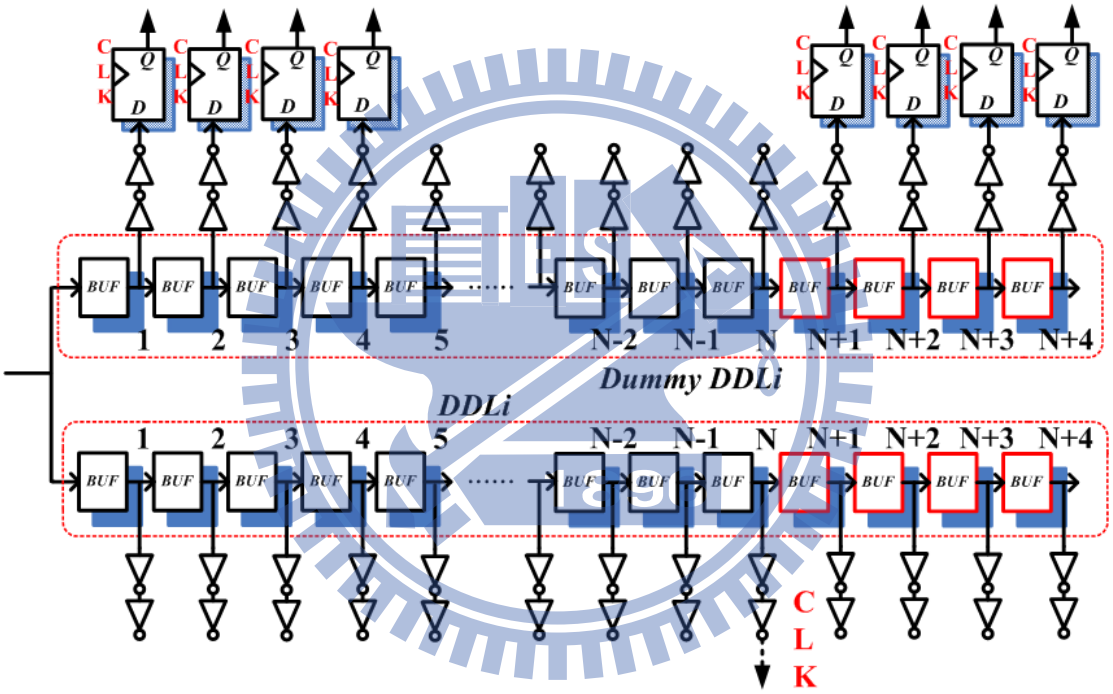


Fig. 3.10 DDLi and dummy DDLi with uniform delay BUF

3.4 The Behavior of Domino AD-SSCG

In this section, we will analyze the behavior for our AD-SSCG structure. Basically, we call the input reference clock of PLL as CLK_REF. This AD-SSCG can be designed for different specifications. There are three main parameters that we can decide. They are (1) frequency of modulated_CLK, (2) modulation frequency, and (3) frequency deviation.

Recalling Fig 3.2 in section 3.2, our Domino modulation method has parameter "N" and "M". It will decide how many clocks in one time modulation. Here, we set parameter "M" is equal to "N" to achieve optimized EMI reduction, the reason will be discussed in section 3.7. The numbers of clocks in one time modulation is (2×N)×N.

In the formula, we define the parameters as follows f_{CLK_REF} is the frequency of CLK_REF, δ is frequency deviation in ppm, and f_m is modulation frequency.

Firstly, we need to estimate how many clocks in one period of modulation cycle. The average frequency of modulated_CLK is $f_{CLK\_REF} \times (1+\frac{1}{2}\delta)$ ($f_{avg} = \frac{f_{max}+f_{min}}{2}$). The time of one period modulation is $1/f_m$. The numbers of clocks (N_C) in one $1/f_m$ period is

$$N_C = \frac{\text{The time of one peroid m}}{\text{The average period of modu}} = \frac{1/f_m}{1/f_{CLK\_REF}(1+\frac{\delta}{2})} = \frac{1}{f_m} \times f_{CLK\_REF}(1+\frac{\delta}{2}) \quad (3.4)$$

Secondly, The numbers of clocks (Nc) in one period of modulation is (2×N)×N.

$$N_C = (2\times N)\times N \quad (3.5)$$

Let equation (3.4) is equal to equation (3.5), and then we can derive the parameter N.

For example, our AD-SSCG is designed to meet SATA-2 specification. The frequency modulation is 30KHz to 33KHz. The frequency deviation is 5000ppm, and it is down spread mode. The frequency of CLK_REF and modulaied_CLK is 100MHz and 99.5MHz respectively. We use the middle value of modulation frequency which is 31.5KHz. We decide these parameters in equation (3.4) and (3.5), and derive that the parameter "N" is 40.

3.5 Specifications of Domino AD-SSCG

Our goal is to generate a 100MHz modulated_CLK with 5000ppm frequency deviation. We use 50ps T_{BUF} to construct the DDLi. Besides 5000ppm frequency deviation, we can also achieve 10000ppm and 15000ppm frequency deviation. The method of implementation is that the amount of phase switching is 2 and 3, it will make the effective T_{BUF} is 100ps and 150ps. There are two kinds of modulated clocks which are narrower and wider modulated clocks in our Domino modulation method. So, we have up and down spread spectrum mode. Besides spread spectrum modes, AD-SSCG can also be operated in non-spread spectrum mode by fixing the phase of DDLi. The DDLi can be regarded as a delay cell. The output waveform of AD-SSCG is the same as CLK_REF.

To summarize, there are totally 7 modes in the proposed Domino AD-SSCG as shown in Table 3.2.

Table 3.2 The specification of the proposed Domino AD-SSCG

Frequency of modulated_CLK	**Spread spectrum mode**	**Frequency deviation**
100MHz	down	5000ppm
100MHz	down	10000ppm
100MHz	down	15000ppm
100MHz	up	5000ppm
100MHz	up	10000ppm
100MHz	up	15000ppm
100MHz	non	0ppm

3.6 The Behavioral Simulations of AD-SSCG and 1.2GHz PLL

3.6.1 Behavioral Model of AD-SSCG

AD-SSCG generates modulated_CLK as the CLK_REF of PLL and Fig. 3.11 (a) is an overall model. Fig 3.11 (b) is AD-SSCG model.

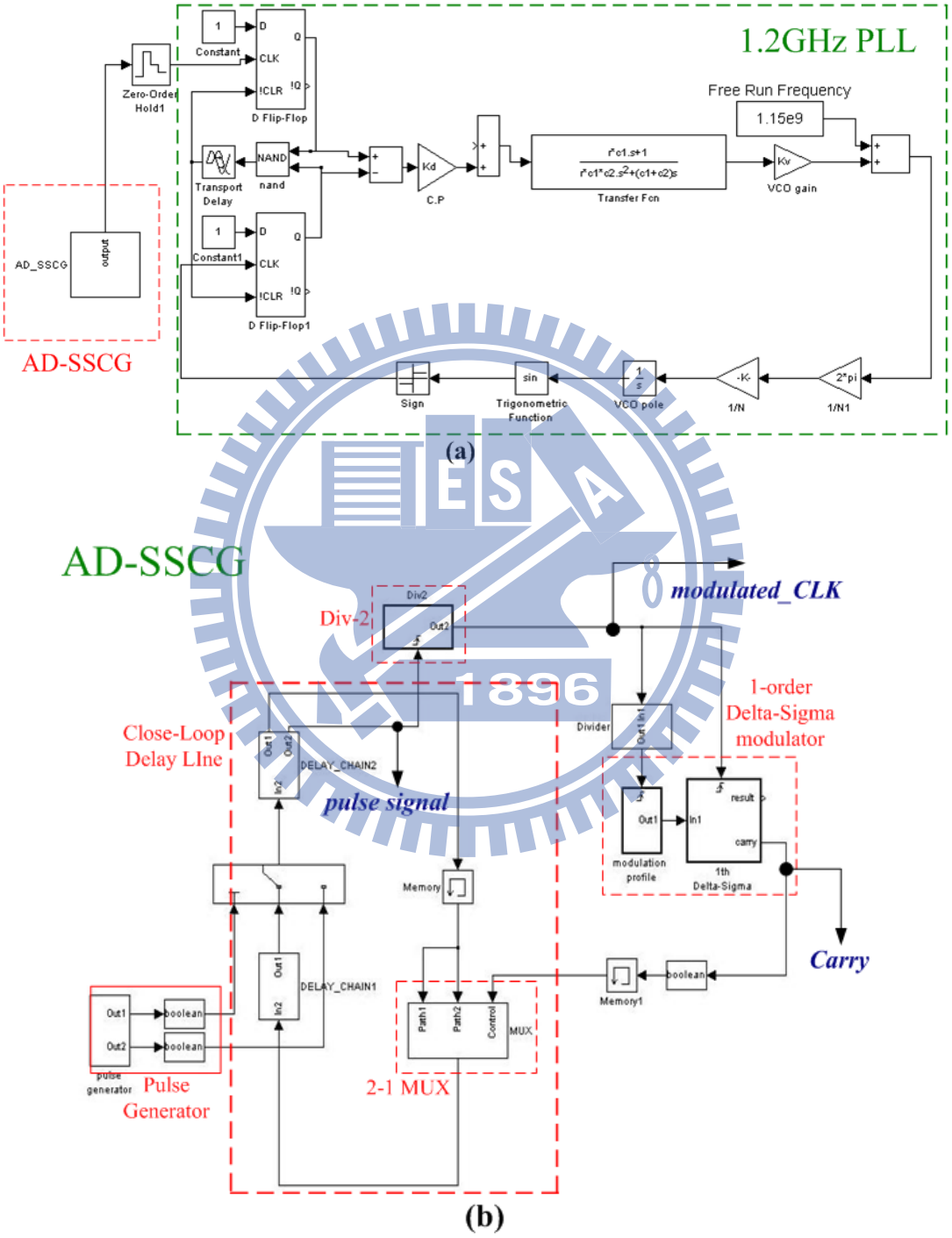


Fig. 3.11 (a) Simulink model of AD-SSCG and PLL
(b) Simulink model of AD-SSCG

We introduce the basic mechanism of AD-SSCG model. The "Pulse Generator" generates a pulse signal, and this pulse signal will go into the "Close-Loop Delay Line". The delay time of "Close-Loop Delay Line" is half period of CLK_REF. The "Div-2" will be triggered two times by pulse signal, and generates a clock whose period is T_{CLK_REF} as shown in Fig. 3.12. We can adjust the delay time of "Close-Loop Delay Line" block to generate wider or narrower modulated clock The "2-1 MUX" in "Close-Loop Delay Line" provides two different paths which have different delay time. When "Carry" of Delta-Sigma modulator is "0", the pulse signal will run the original path whose delay time is $T_{CLK\_REF}/2$ in "Close-Loop Delay Line". But when "Carry" of Delta-Sigma modulator is "1", the pulse signal will run anther path which is "Path2" of "2-1 MUX", and the delay time is $T_{CLK\_REF}/2 + T_{MUX}$. If the additional T_{MUX} is positive, the delay time of "Close-Loop Delay Line" is increased to generate wider modulated clock. And if the additional T_{MUX} is negative, the delay time of "Close-Loop Delay Line" is decreased to generate narrower modulated clock. Fig. 3.13 shows modulated_CLK with one wider modulated clock when "Carry" is "0" and "1".

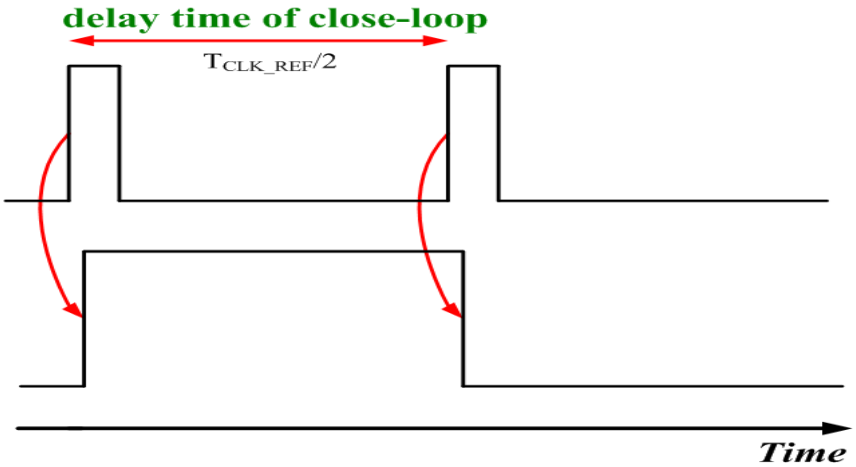


Fig. 3.12 Two pulse signals to generate a modulated clock

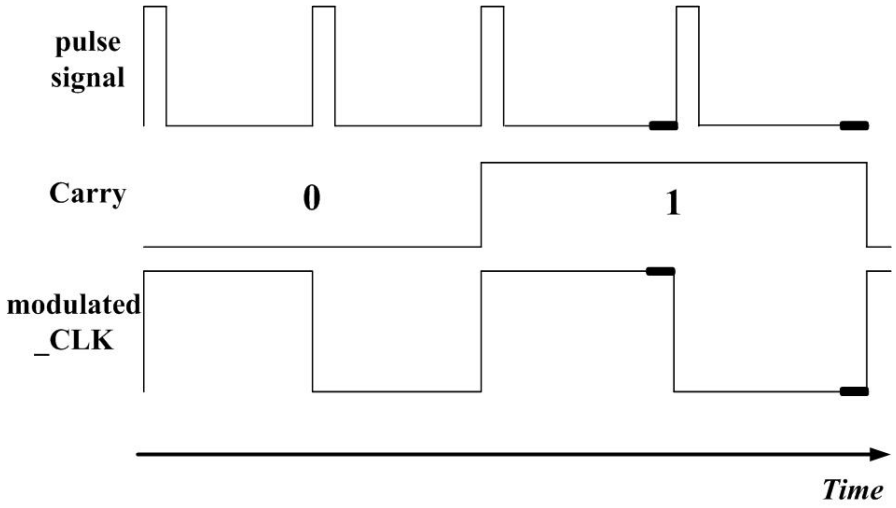


Fig. 3.13 The time diagram of modulated_CLK when "Carry=0 and 1"

The parameters of PLL are provided by out previous design [13]. Table 3.3 shows all the parameters we use to do the simulation. The key specifications of PLL also listed in Table 3.3.

Table 3.3-1 The parameters of 1.2GHz PLL [13]

K_{VCO}		700MHz/V
I_p		50 uA
loop filter parameters	C_1	70pf
	C_2	4.6pf
	R_1	4.5kΩ
	R_2	1.8 kΩ
	C_3	1.8pf
N(divider ratio)		12
Input Frequency		100MHz
Output Frequency		1.2GHz

Table3.3-2 The simulation results of 1.2GHz PLL [13]

Items	PLL	
Technology	UMC 90nm 1P9M	
Power Supply	1V	
Crystal Frequency	100MHz	
VCO tuning frequency	1.3~1.5GHz	
K_{VCO}	670MHz/V	
Settling time	<3us	
Jitter performance	σ_{RMS}	810fs
	σ_{p2p}	3.88ps
Power consumption	5.87mW	
Core Area	Main circuit	$170\times80um^2$
	Loop Filter	$235\times325um^2$

3.6.2 Down Spread Mode

Fig. 3.14 shows the non-spread spectrum mode to make sure the locking ability of PLL. Fig. 3.15, Fig.3.16 and, Fig.3.17 are respectively 5000ppm, 10000ppm and, 15000ppm with down spread mode. The EMI reduction with 5000ppm, 10000ppm, and 15000ppm are separately 18.0dB, 21.0dB, and 21.5dB.

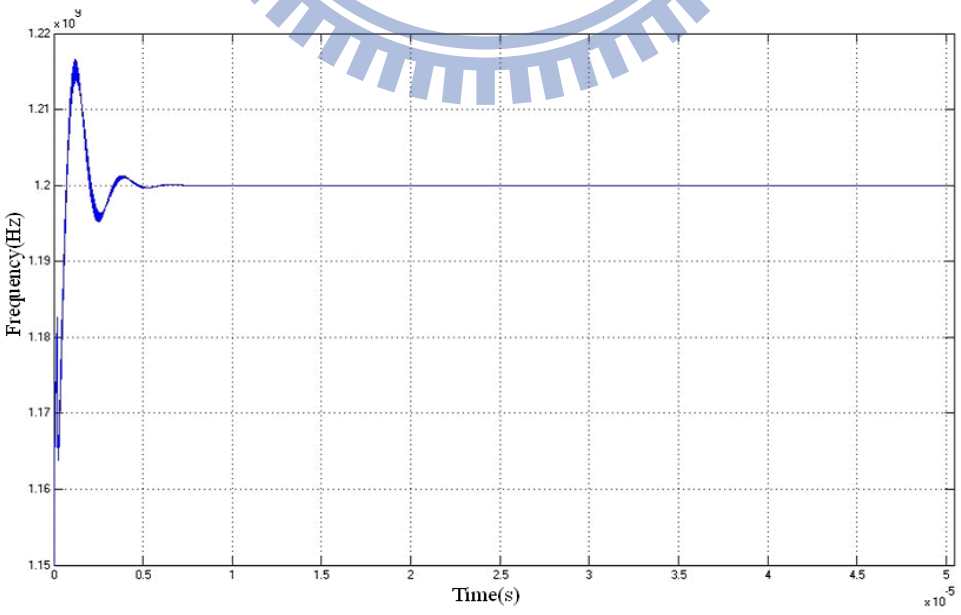


Fig. 3.14 Lock-in behavior of 1.2GHz VCO

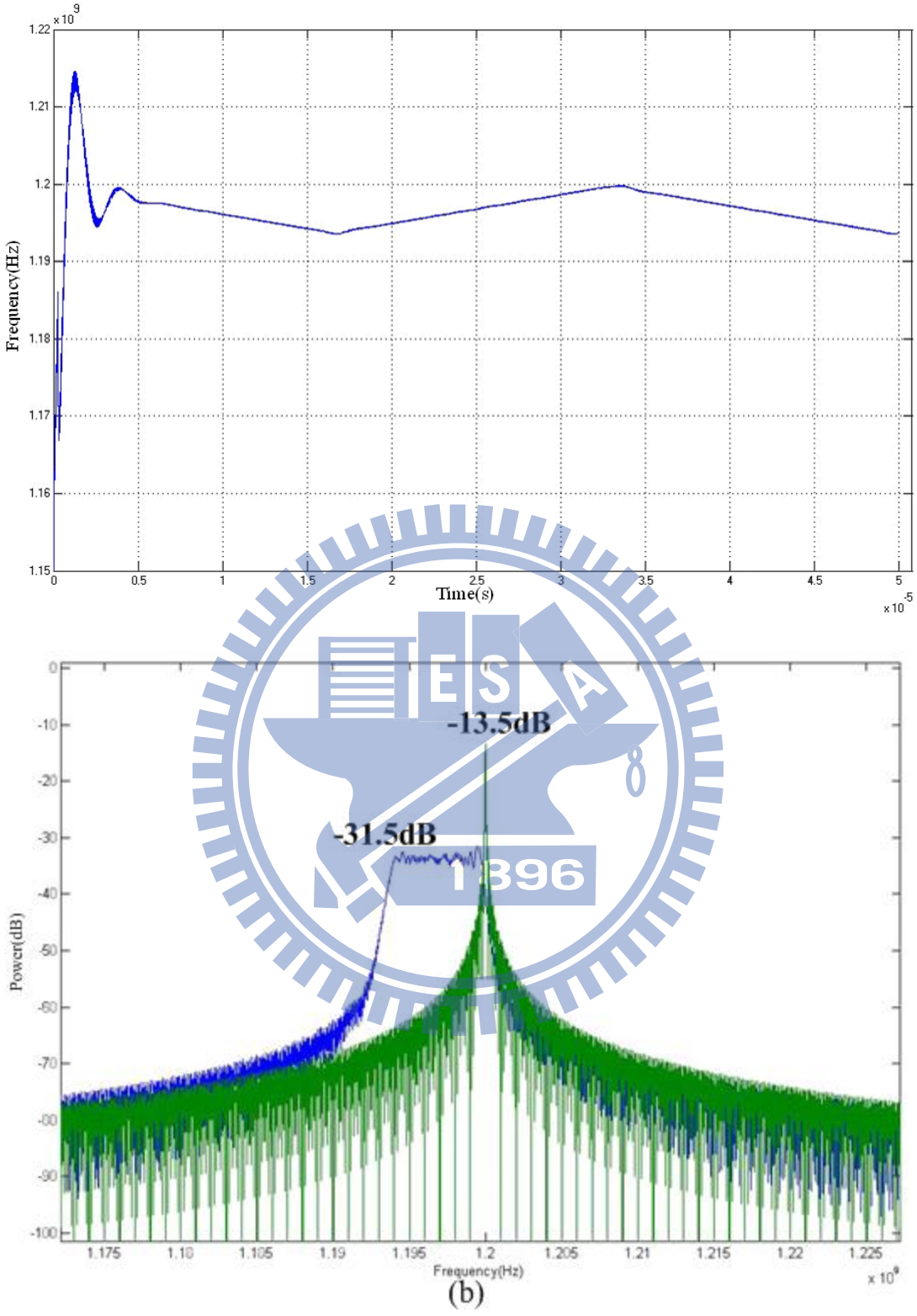


Fig. 3.15 (a) Frequency of PLL with 5000ppm and down mode
(b) Spectrum of PLL with 5000ppm and down mode

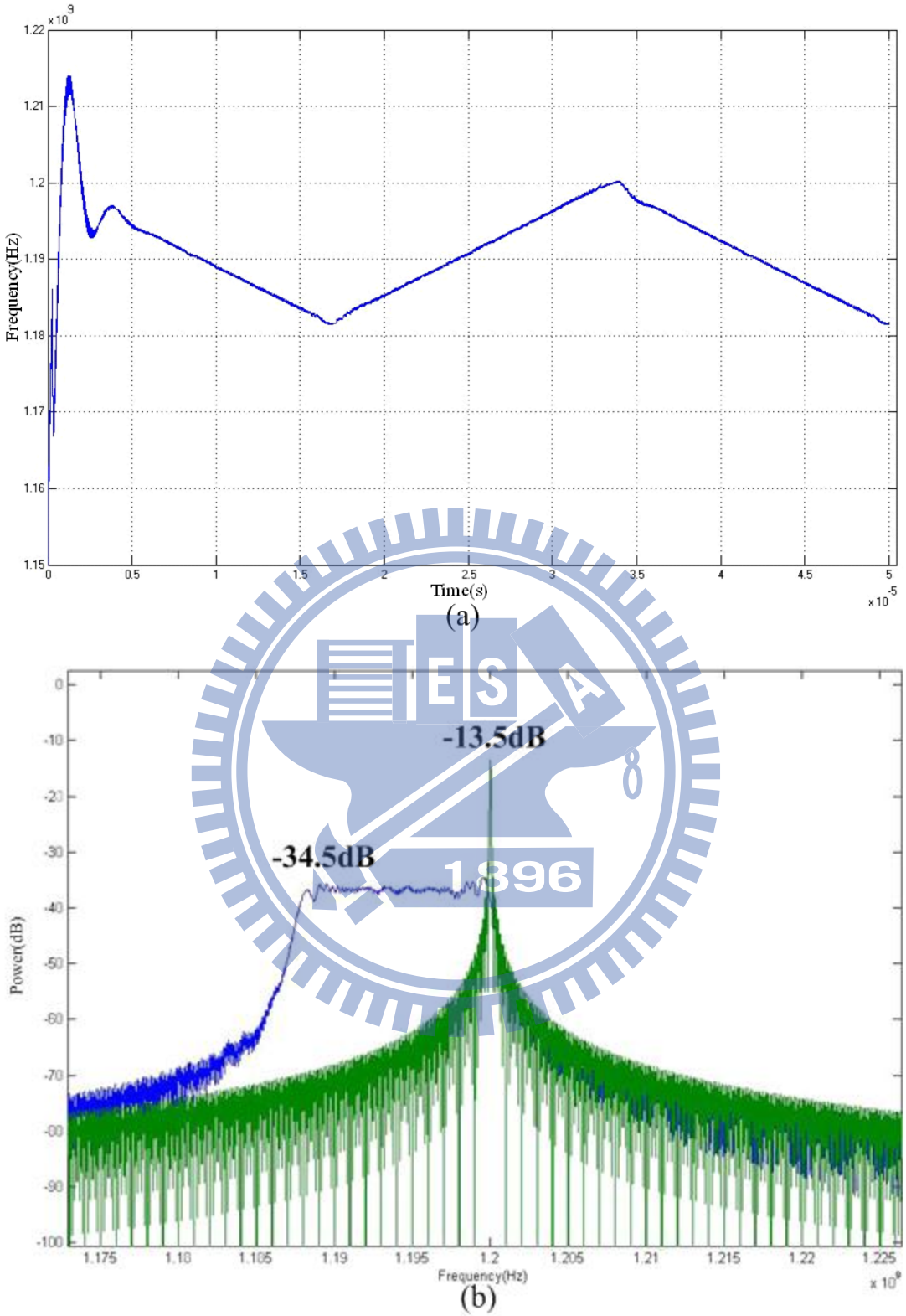


Fig. 3.16 (a) Frequency of PLL with 10000ppm and down mode
(b) Spectrum of PLL with 10000ppm and down mode

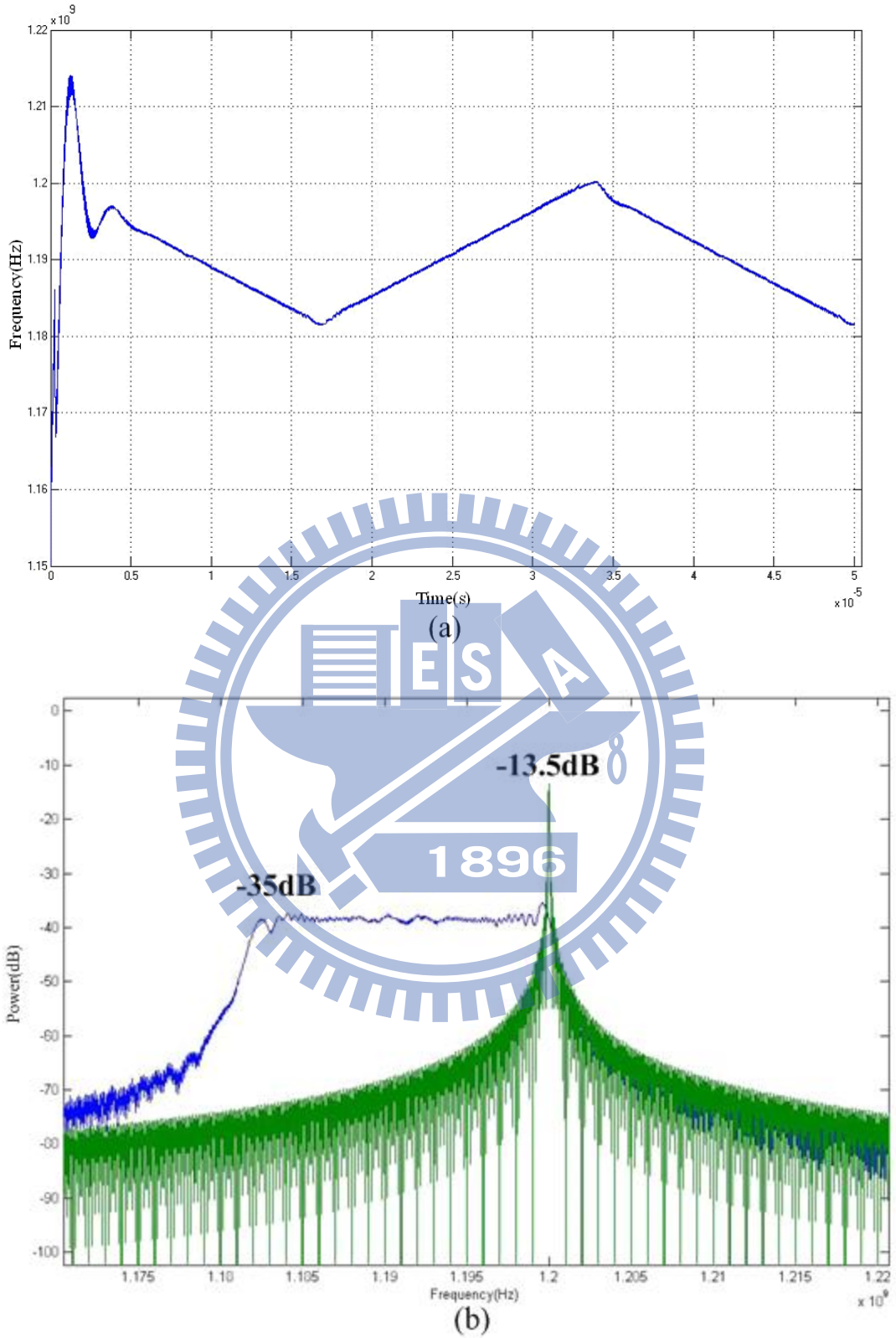


Fig. 3.17 (a) Frequency of PLL with 15000ppm and down mode
(b) Spectrum of PLL with 15000ppm and down mode

3.6.3 Up Spread Mode

Fig. 3.18, Fig.3.19 and, Fig.3.20 are respectively 5000ppm, 10000ppm and, 15000ppm with up spread mode. The EMI reduction with 5000ppm, 10000ppm, and 15000ppm are respectively 18.1dB, 20.0dB, and 21.7dB.

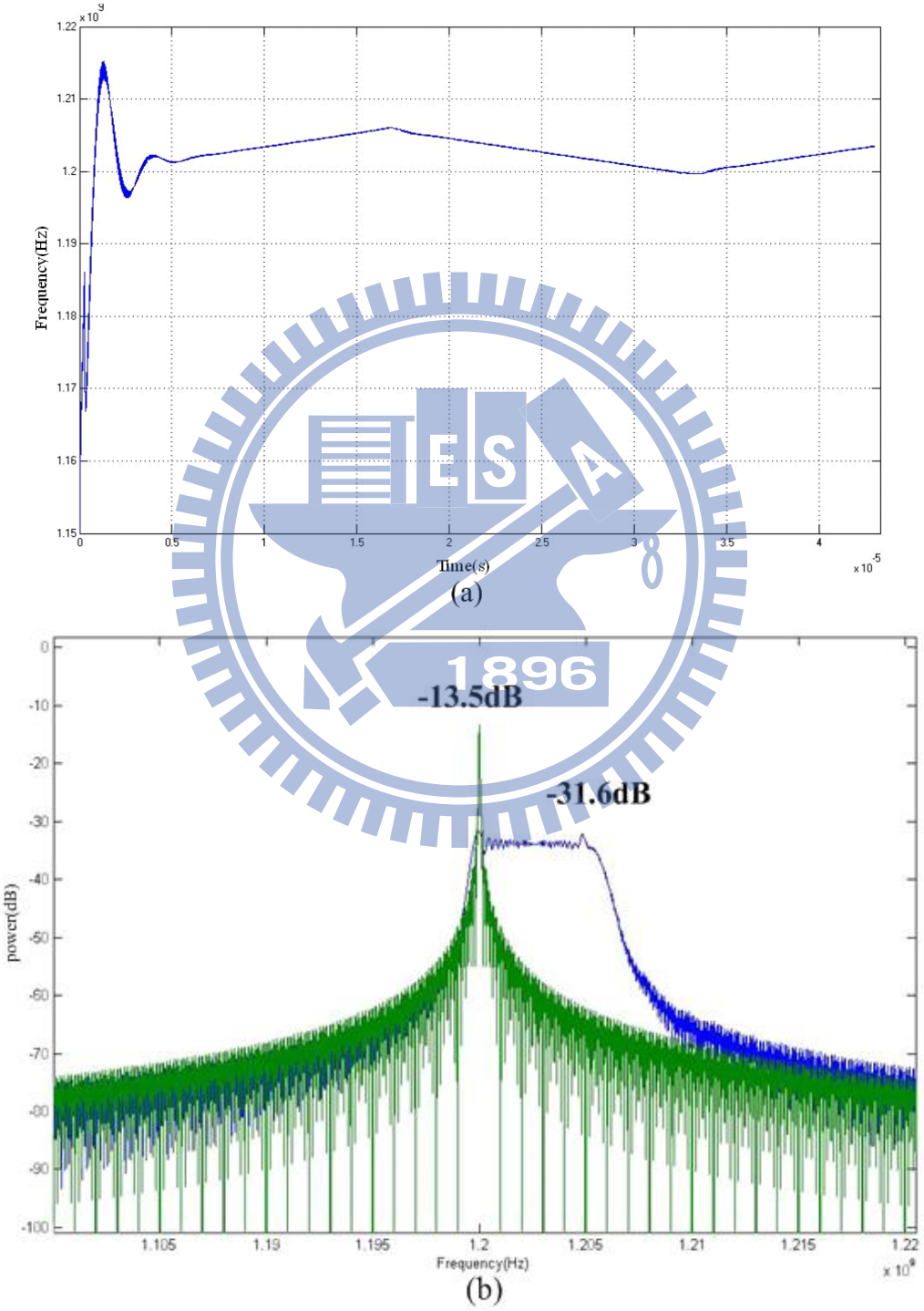


Fig. 3.18 (a) Frequency of PLL with 5000ppm and up mode
(b) Spectrum of PLL with 5000ppm and up mode

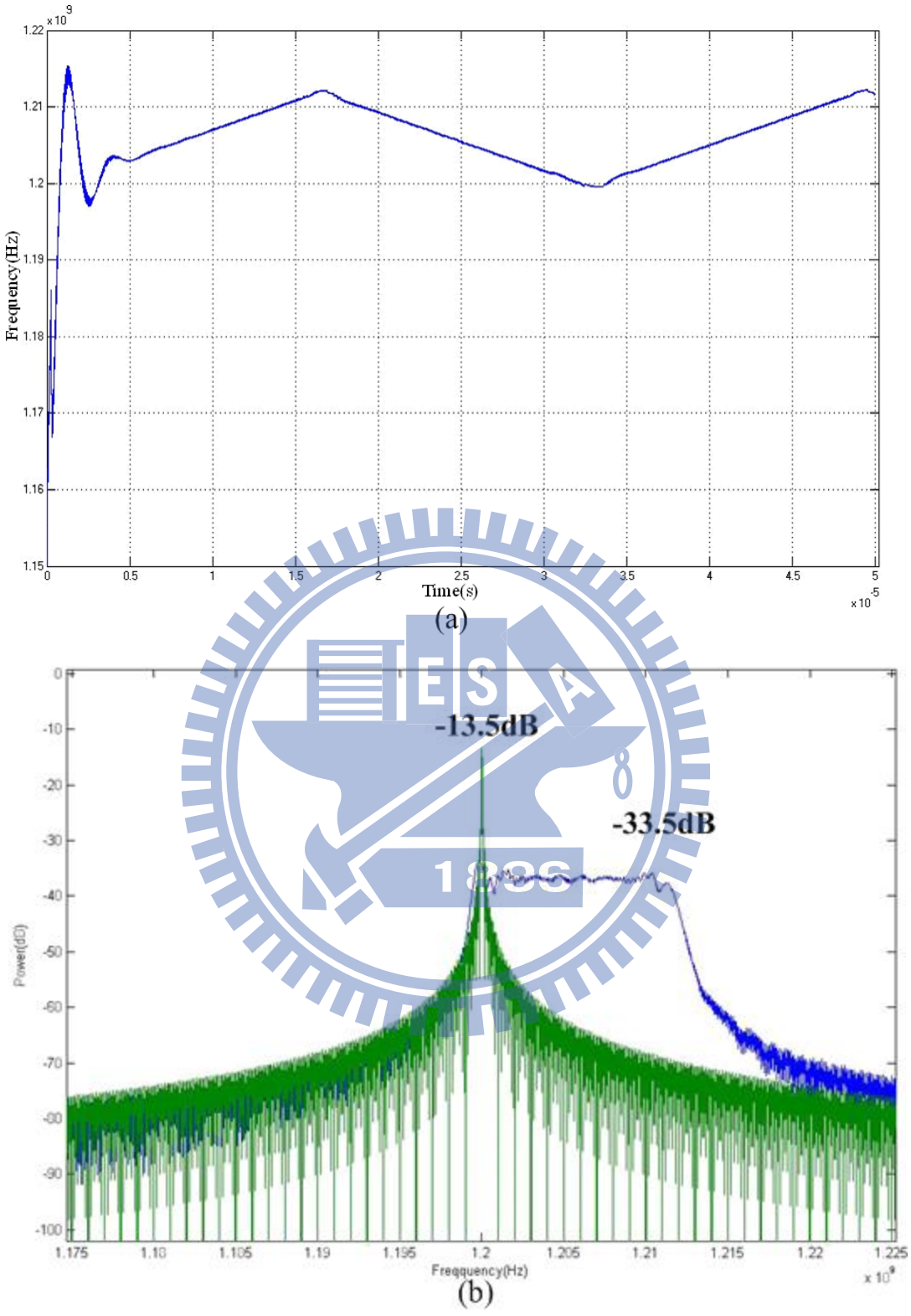


Fig. 3.19 (a) Frequency of PLL with 10000ppm and up mode
(b) Spectrum of PLL with 10000ppm and up mode

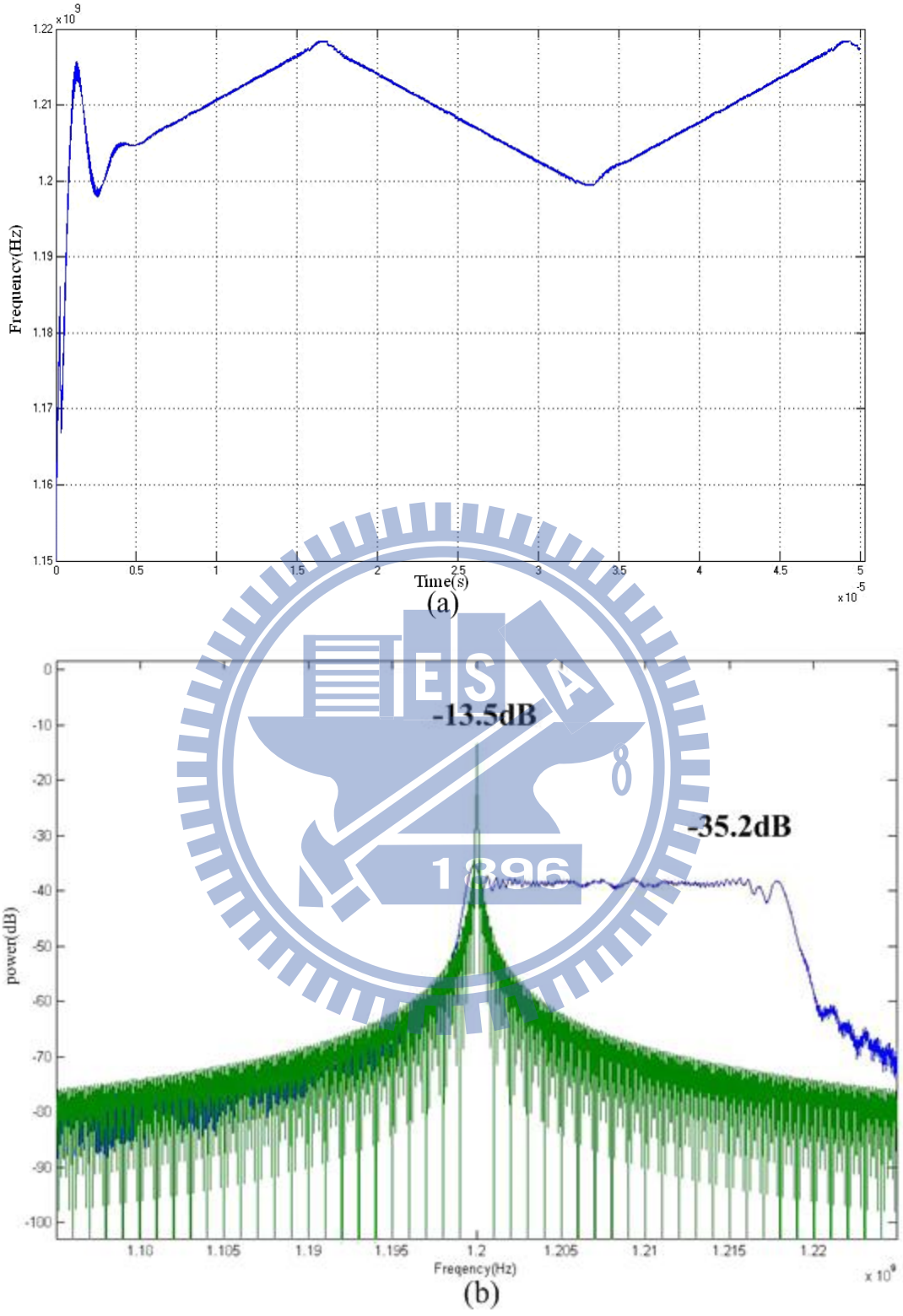


Fig. 3.20 (a) Frequency of PLL with 15000ppm and up mode
(b) Spectrum of PLL with 15000ppm and up mode

3.7 The Parameters Optimization of Domino AD-SSCG

In this section, we will discuss the reason that we set parameter "M=N" to achieve optimized EMI reduction. Here, N defines how many clock in a group, and M defines how many groups in the half of the modulation cycle. We will discuss the condition of that when (1) "N=M", (2)"N>M", and (3) "N<M". Then, we will find the sequence of (2) and (3) will be the same. So, we just compare the condition of N=M and N>M. We will show a simple example when N is equal to different M, and we will get rough understanding of different modulation scheme. Then, we use simulink model to verify our thinking to make sure the "N=M" is the best choice for this modulation method.

Our modulation method is that we use two kinds of clock to generate modulated_CLK. The first group of all clocks is derived from CLK_REF. And when the modulated_CLK has maximum frequency deviation, all clocks in that group is replaced by modulated clocks. The average frequency of that group is $f_n + \delta f_n$. The modulation method of (a) "N<M", (b)"N=M", and (c) "N>M" is shown in Fig. 3.21. We assume N×M=16 to simply the plot. Then, we combine the first four rows and the second four rows together of modulation scheme (a), and the sequence of modulated_CLK is shown in Fig. 3.22. It will shows that the modulation scheme (a) and (c) have one same feature of that there are four clocks will be replaced in the first group and eight clocks will be replaced in the second group, but the modulated clocks will appear at different positions between (a) and (c). But when we implement the AD-SSCG, we will use 1^{st}-ΣΔ modulator to randomize the modulated clock. Thus, the modulated_CLK of (a) and (c) will be the same, that is, the modulated clocks will appear in the same positions. So, we only need to compare the modulation scheme when (a) "N=M" and (b) "N>M".

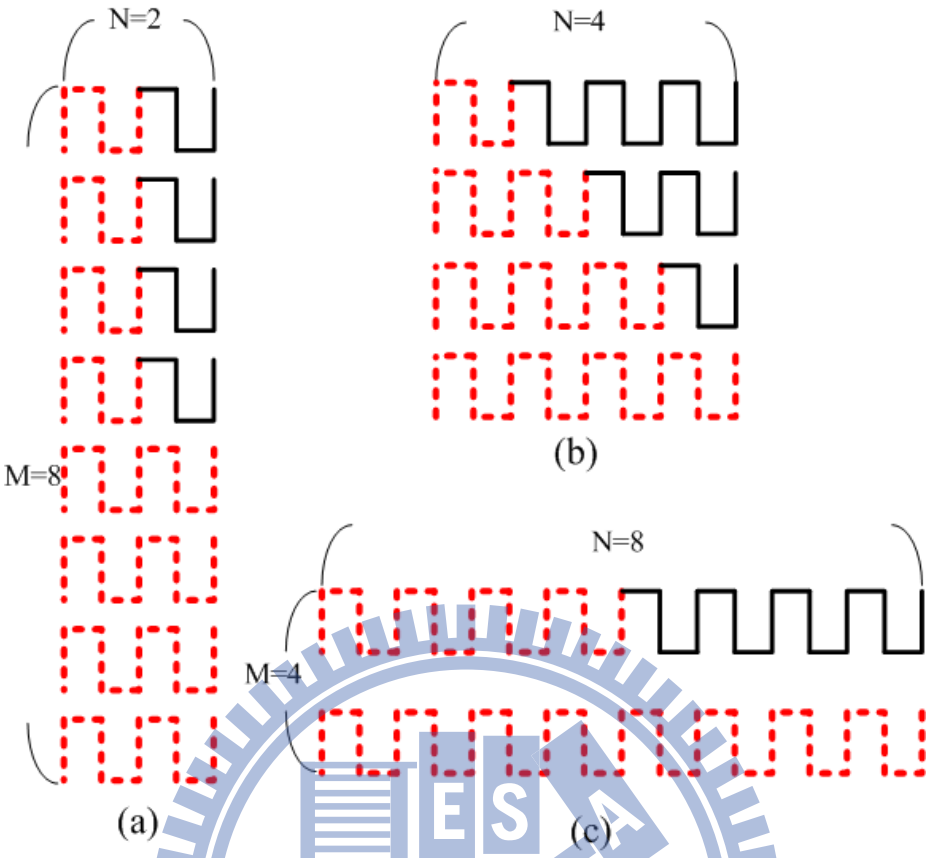


Fig. 3.21 The modulation scheme of different N and M.
(a) N=2 M=8 (N<M) (b) N=4 M=4 (N=M) (c)N=8 M=2 (N>M)

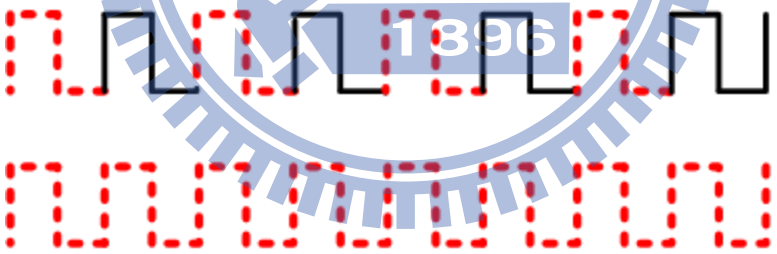


Fig. 3.22 Re-combination of modulation_CLK of (a) N=2 M=8.

Here, we assume N×M =64 to show three different modulation methods. In our AD-SSCG, the "N×M =1600", it is difficult to explain the behavior with different modulation methods. So we set N×M =64 as a simple example to explain here. They are respectively (a) N=M=8, (b) N=16 M=4, and (c) N=32 M=2 as shown in Fig. 3.23. We can observe the average frequency of the first group of modulation scheme (a) is $f_n + \frac{1}{8}\delta f_n$, then the second group is $f_n + \frac{2}{8}\delta f_n$, and so on. We observe the average

frequency of the first group of modulation scheme (b) is $f_n + \frac{1}{4}\delta f_n$, then the second is $f_n + \frac{2}{4}\delta f_n$, and so on. We observe the average frequency of modulation scheme (c) the first group is $f_n + \frac{1}{2}\delta f_n$, and then the second group is $f_n + \frac{2}{2}\delta f_n$. The difference of modulation scheme is that the step numbers in frequency deviation. If there are more steps, the modulation profile will be more like ideal triangular profile. When PLL tracks the modulated_CLK, the spread frequency of PLL will be similar to ideal triangular profile. The peak power of PLL will be dissipated effectively on the spread spectrum bandwidth. Here, we think when "N=M" will be the best choice for EMI reduction because the "N=M" modulation scheme have the most steps in maximum frequency deviation. The next paragraph will use simulink model to verify further.

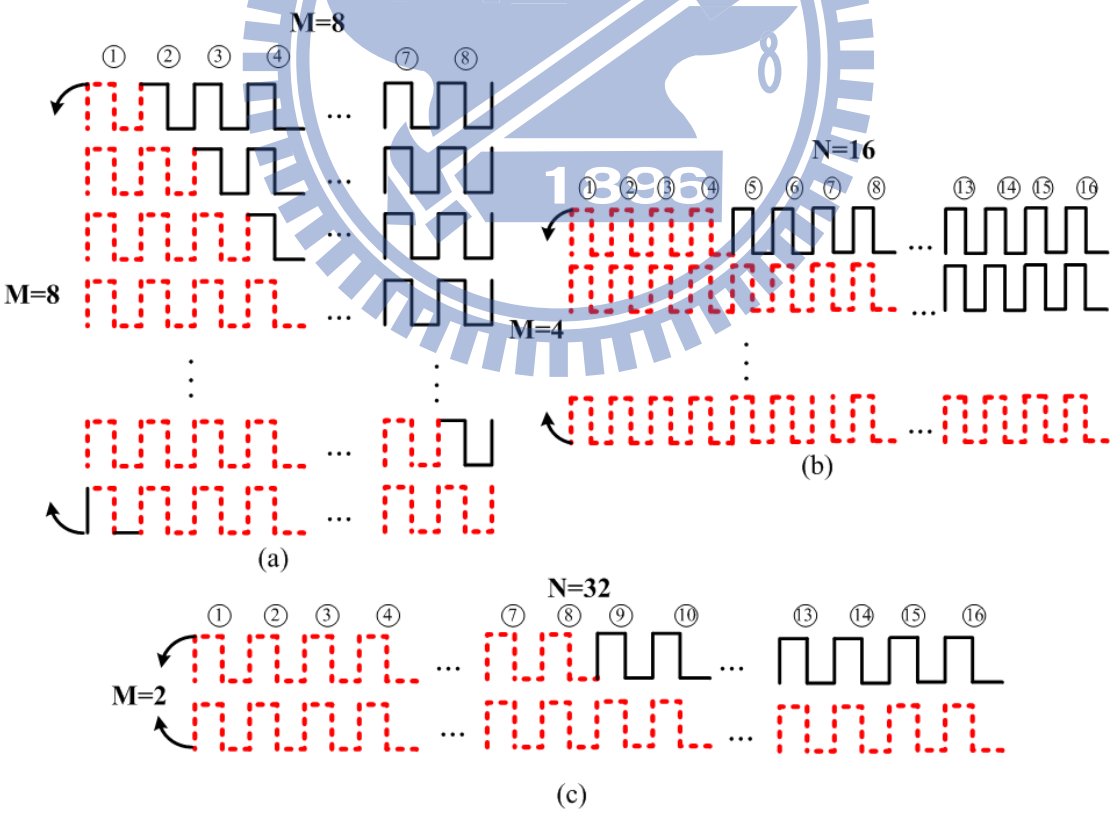
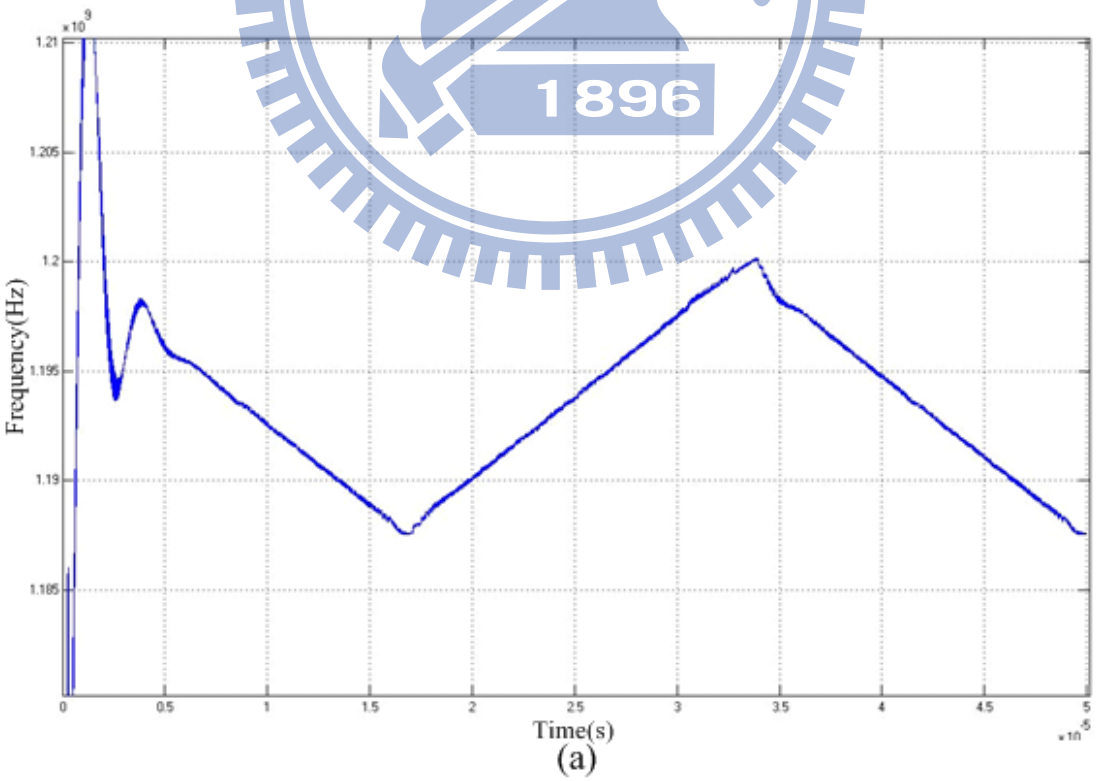


Fig. 3.23 The modulation scheme of assuming N×M =64.
(b) N=8 M=8 (b) N=16 M=4 (c)N=32 M=2

Fig. 3.24 and Fig. 3.25 shows simulation results that we use the same behavior

model of AD-SSCG and PLL in section 3.6 but with different setting of parameters N and M. In this simulation, AD-SSCG is operated in down spread mode with 10000ppm frequency deviation. In our design, "N×M = 1600", and three kinds of modulation scheme are used respectively (a) N=M=40, (b) N=80 M=20, and (c) N=160 M=10. Firstly, Fig. 3.24 shows modulation profile (a) is better than (b). Moreover, in the modulation scheme of (c), the triangular modulation profile is distorted severely because the steps of numbers in frequency deviation are not enough. Secondly, the EMI reduction of these three schemes is shown in Fig. 3.25, they are respectively (a) 21.5dB, (b) 21.5dB, and (c) 18.5dB. According to the verification of simulink model, the parameter of "N"and "M" in the proposed Domino modulation method should be the same to achieve the optimized EMI reduction.



(a)

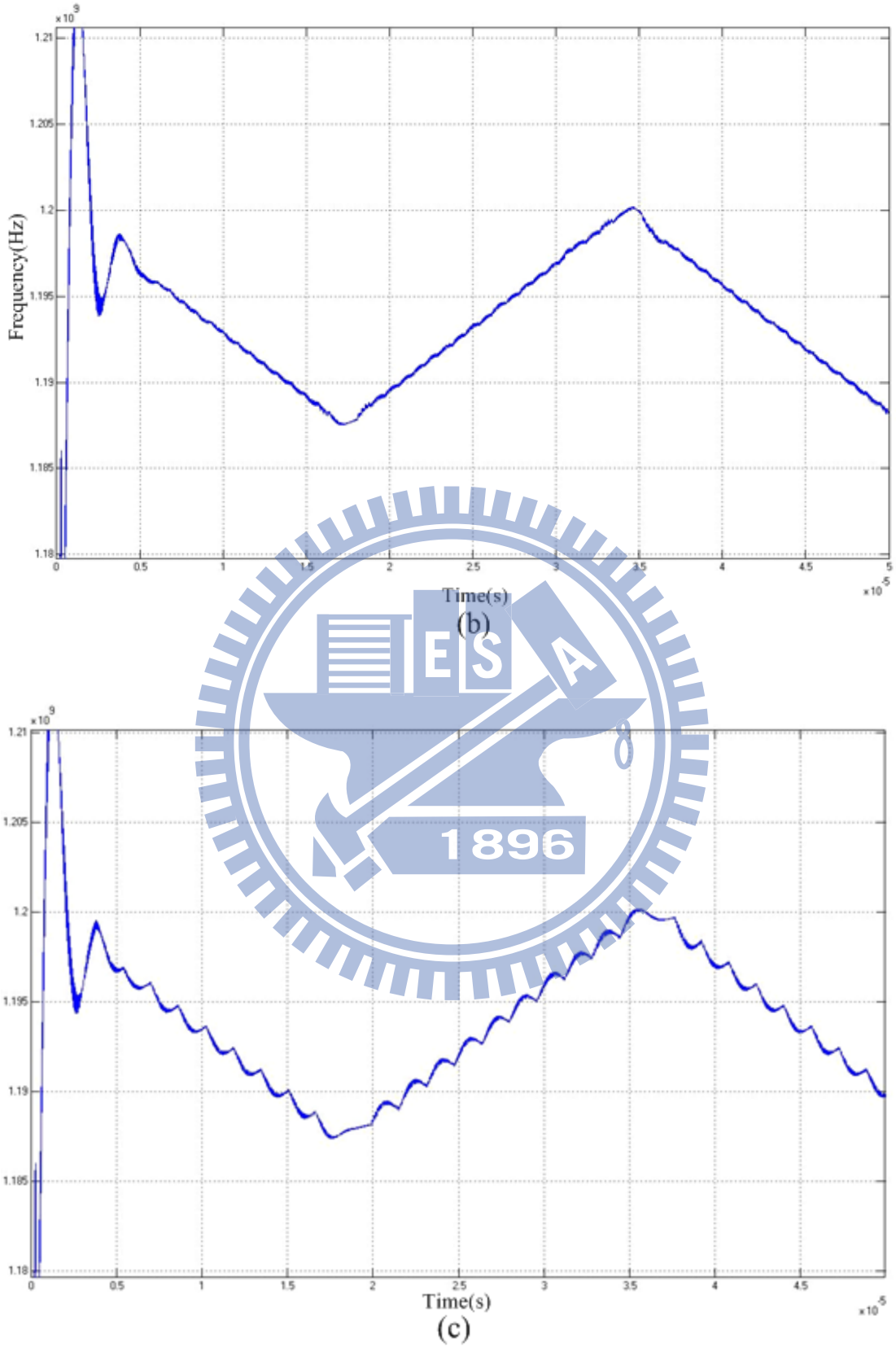
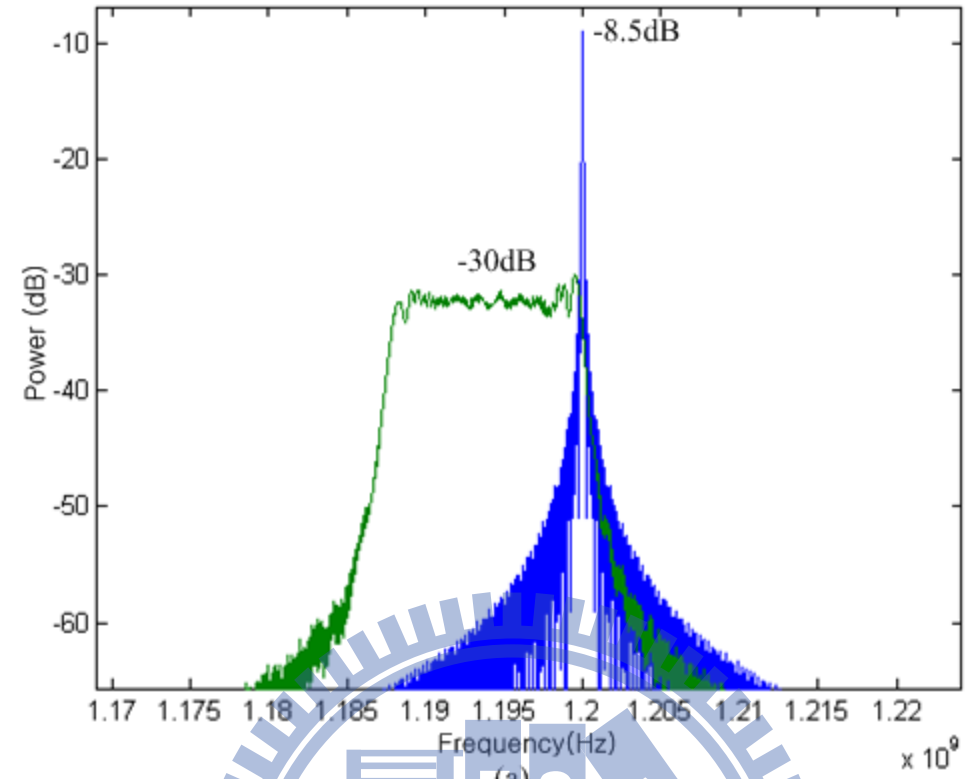
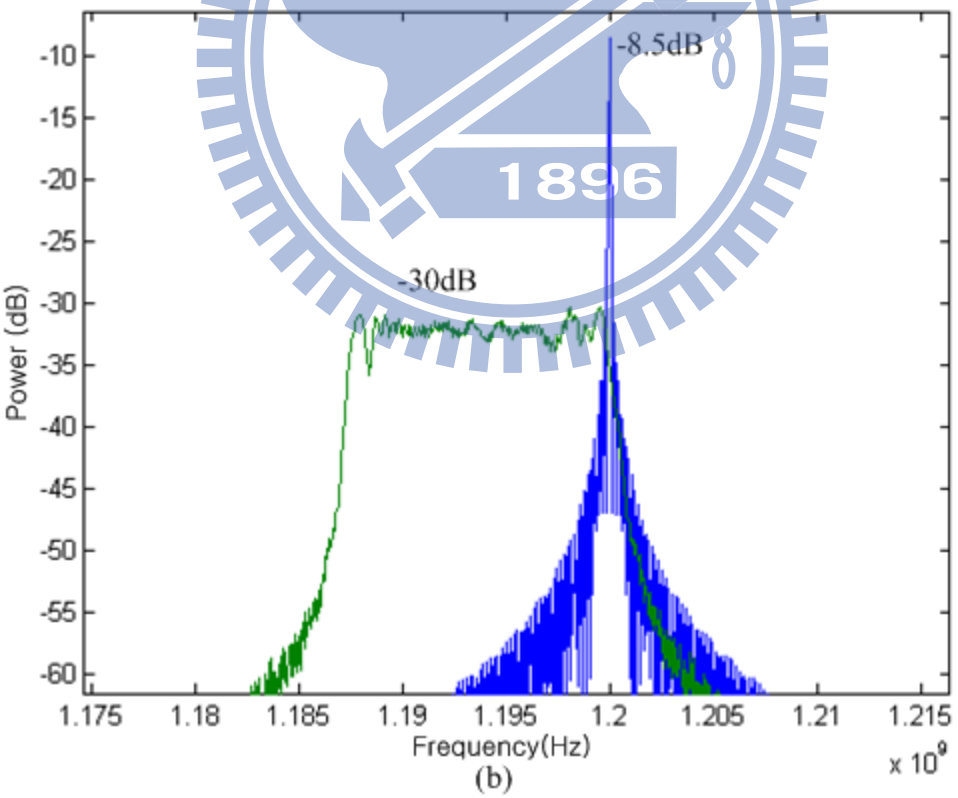


Fig. 3.24 Frequency of PLL with down spread mode and 10000ppm frequency deviation (a) N=M=40 (b) N=80 M=20 (c) N=160 M=10


-8.5dB
-30dB
Power (dB)
-10
-20
-30
-40
-50
-60
1.17
1.175
1.18
1.185
1.19
1.195
1.2
1.205
1.21
1.215
1.22
Frequency(Hz)
x 10^9

(a)


-8.5dB
-30dB
Power (dB)
-10
-15
-20
-25
-30
-35
-40
-45
-50
-55
-60
1.175
1.18
1.185
1.19
1.195
1.2
1.205
1.21
1.215
Frequency(Hz)
x 10^9

(b)

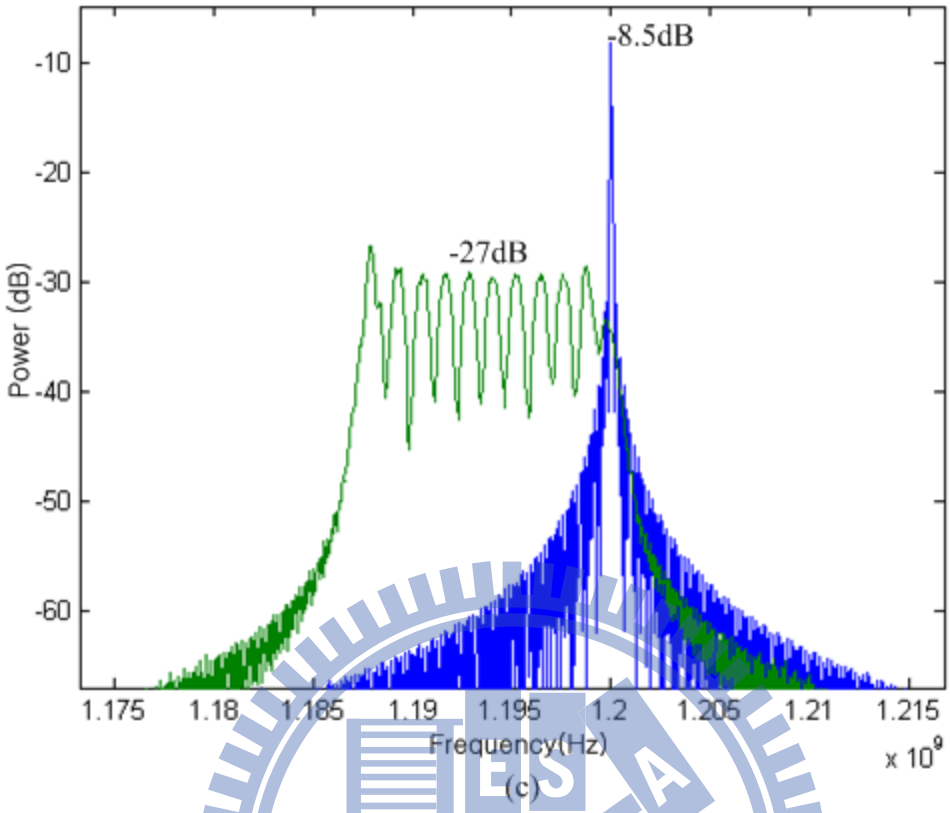


Fig. 3.25 Spectrum of PLL with down spread mode and 10000ppm frequency deviation (a) N=M=40 (b) N=80 M=20 (c) N=160 M=10

3.8 The Performance Comparison

Here, we want to analyze the SSCG of input reference modulation between JSSC 2007 [12] and ours. The setting parameters are (1) down spread mode, (2) frequency deviation =10000ppm, and (3) modulated_CLK=100MHz. In our design, the numbers of steps in maximum frequency deviation is 40. In paper [12], the numbers of steps in maximum frequency deviation is 6. In order to compare the advantage and disadvantage of these two modulation methods, we use simulink to establish the model of paper [12] as shown in Fig. 3.26 and we also set the numbers of steps in maximum frequency deviation is also 40. However, if we adopt the modulation method of paper [12] to make 40 steps, the delay BUF is 2.5ps (10ps/4 =2.5ps). It ‘s

too small to implement on chip. So, we adopt the modulation method of paper [12] but have reasonable steps of 10. It means we need to make a delay BUF whose delay time is 10ps (1/CLK_REF=10000ps, 10000ps×1%=100ps, 100ps/10=10ps). Totally, there are three experimental results: (a) Paper [12] with 40 steps, and the assuming is that the delay BUF is 2.5ps, it can't be implemented. (b) Domino modulation method with 40 steps. (c) Paper [12] with 10 steps, and the delay BUF may have reasonable delay time which have chance to implement on chip.

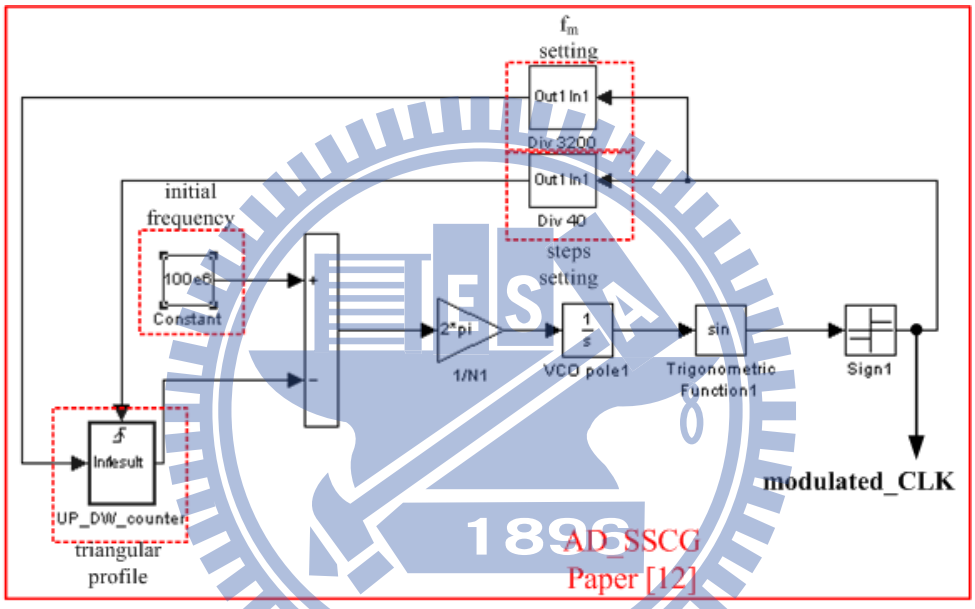
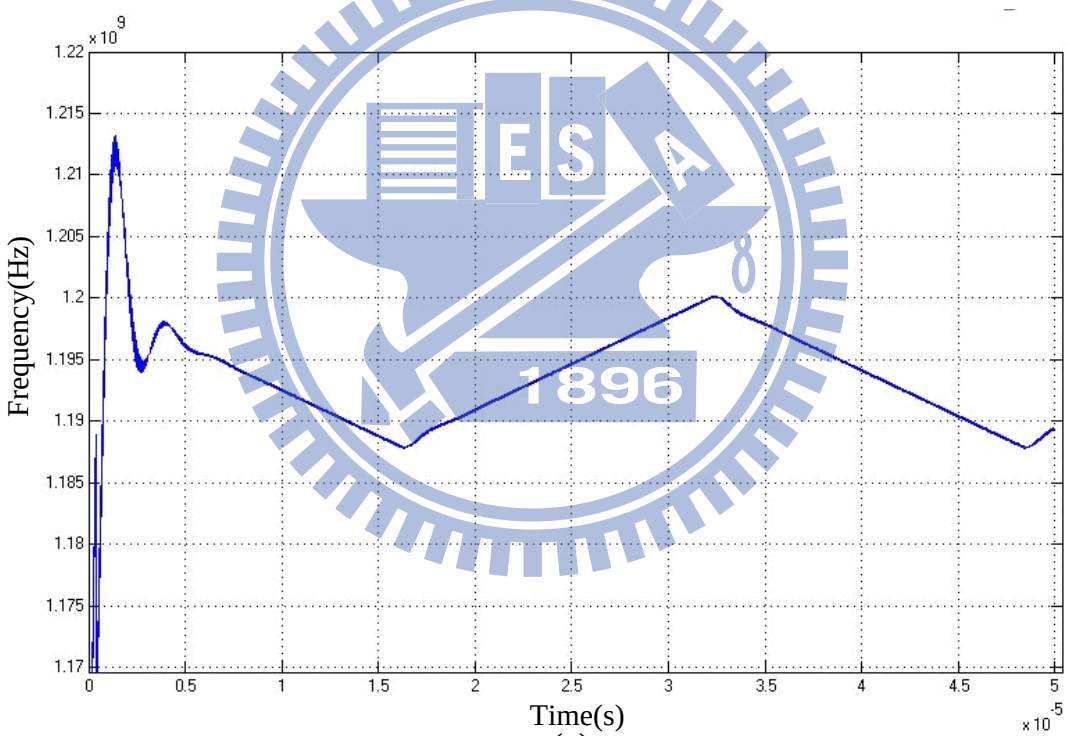


Fig. 3.26 Simulink model of AD-SSCG of paper [12]

In Fig. 3.27, (a) and (b) show no significant difference in triangular profile between these two modulation methods. In Fig. 3.28, we compare the EMI reduction, and (a) and (b) have 22 and 21 dB which is comparable. Although, we don't have better EMI reduction but the proposed Domino modulation method can be implemented on chip because the required delay time of delay BUF is larger than paper [12]. On the other hand, we compare Domino modulation and reasonable modulation method of paper [12]. Firstly, we can see in Fig.3.27, our modulation

profile is more like ideal triangular profile. Secondly, the EMI reduction of (b) and (c) are respectively 21dB and 20dB as shown in Fig. 3.28. Finally, we make the summary of these two methods. If the required modulated_CLK has the feature of lower frequency or larger frequency deviation, we can adopt the modulation method of paper [12] because the EMI reduction is a little better than ours. But if the required modulated_CLK has the feature of higher frequency or less frequency deviation, we recommend our modulation method because the required delay BUF can be implemented on practical chip.



(a)

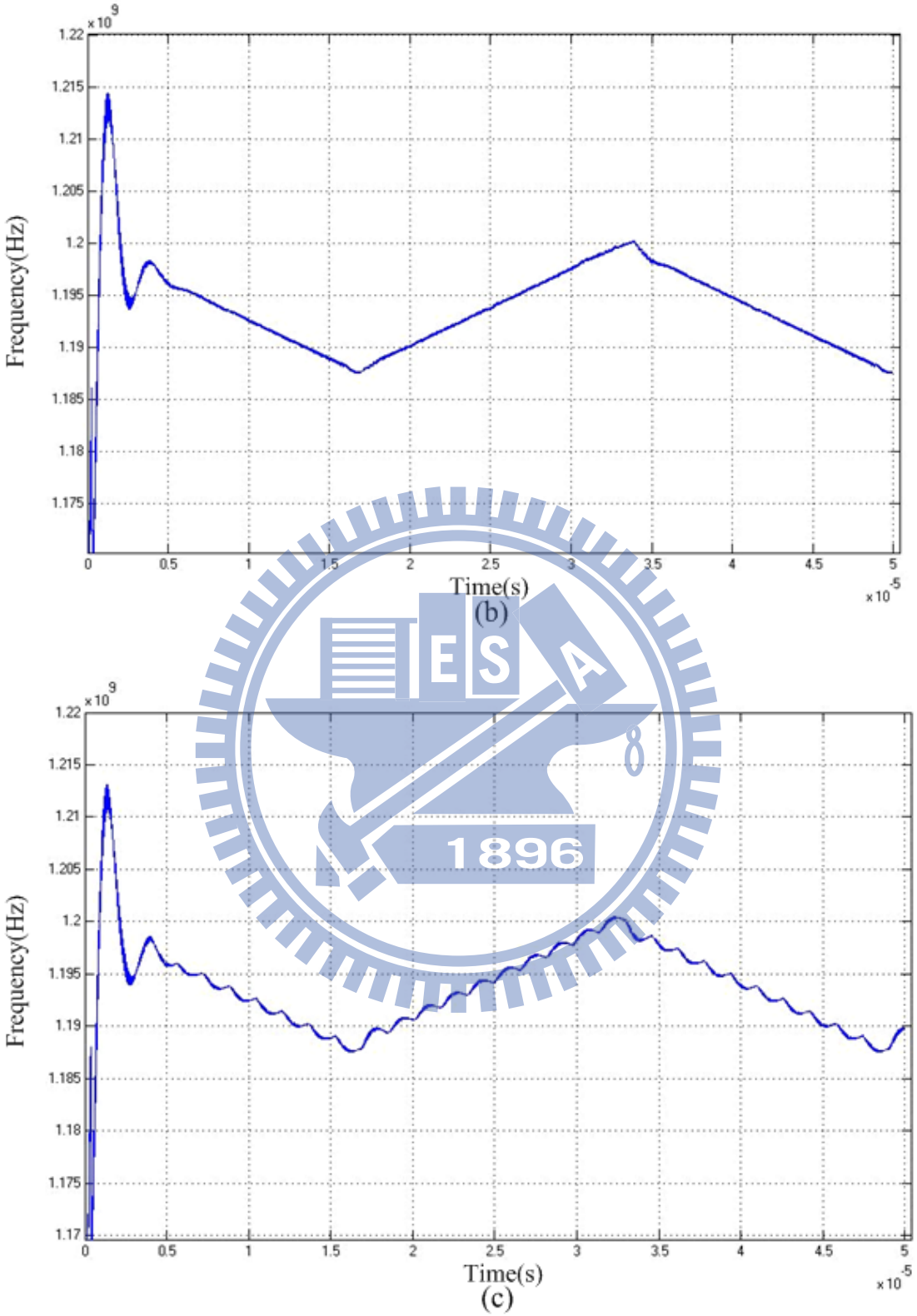
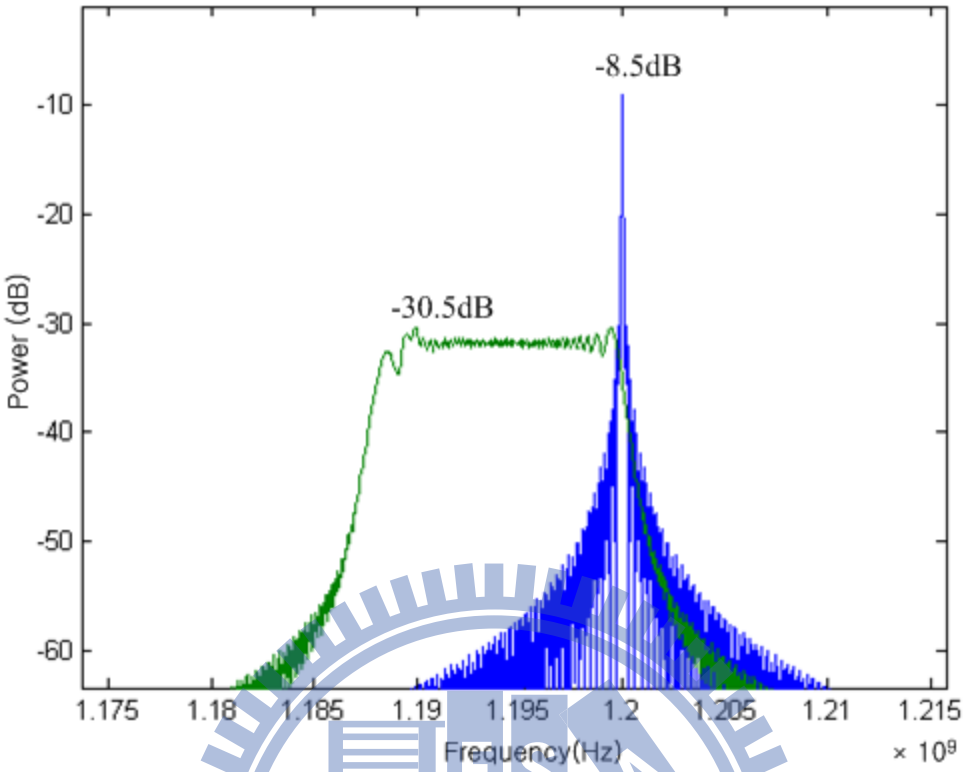
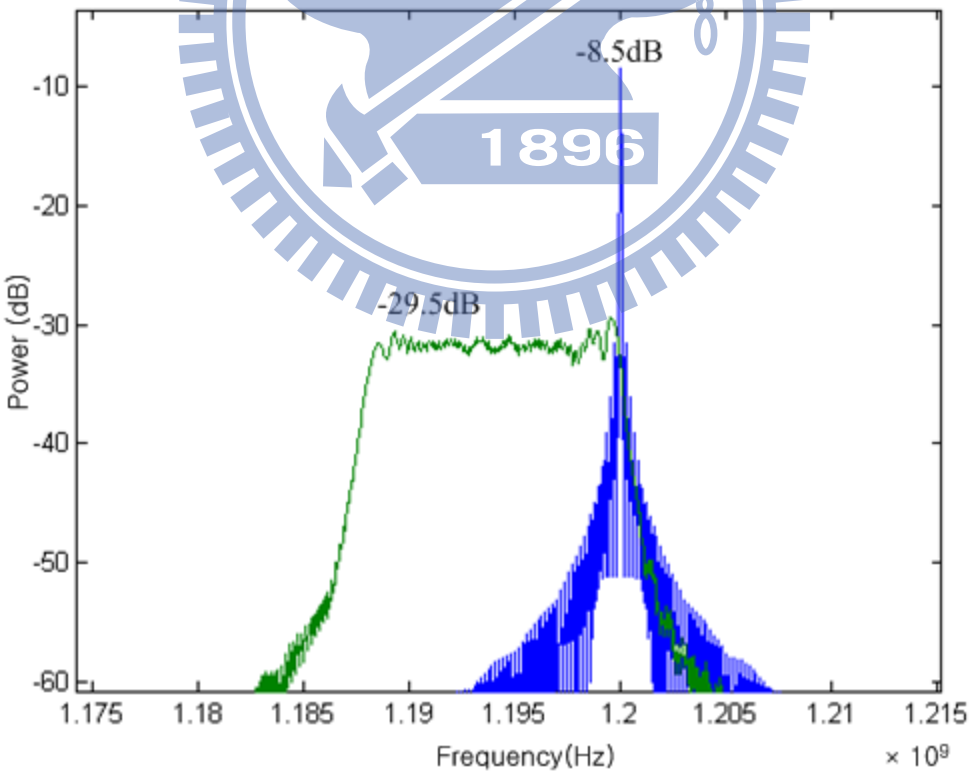


Fig. 3.27 Frequency of PLL with down spread mode and 10000ppm frequency deviation (a) paper [12] with 40 steps (b) proposed method with 40 steps (c) paper [12] with 10 steps


-8.5dB
-30.5dB
Power (dB)
-10
-20
-30
-40
-50
-60
1.175
1.18
1.185
1.19
1.195
1.2
1.205
1.21
1.215
Frequency(Hz)
× 10^9

(a)


-8.5dB
-29.5dB
Power (dB)
-10
-20
-30
-40
-50
-60
1.175
1.18
1.185
1.19
1.195
1.2
1.205
1.21
1.215
Frequency(Hz)
× 10^9

(b)

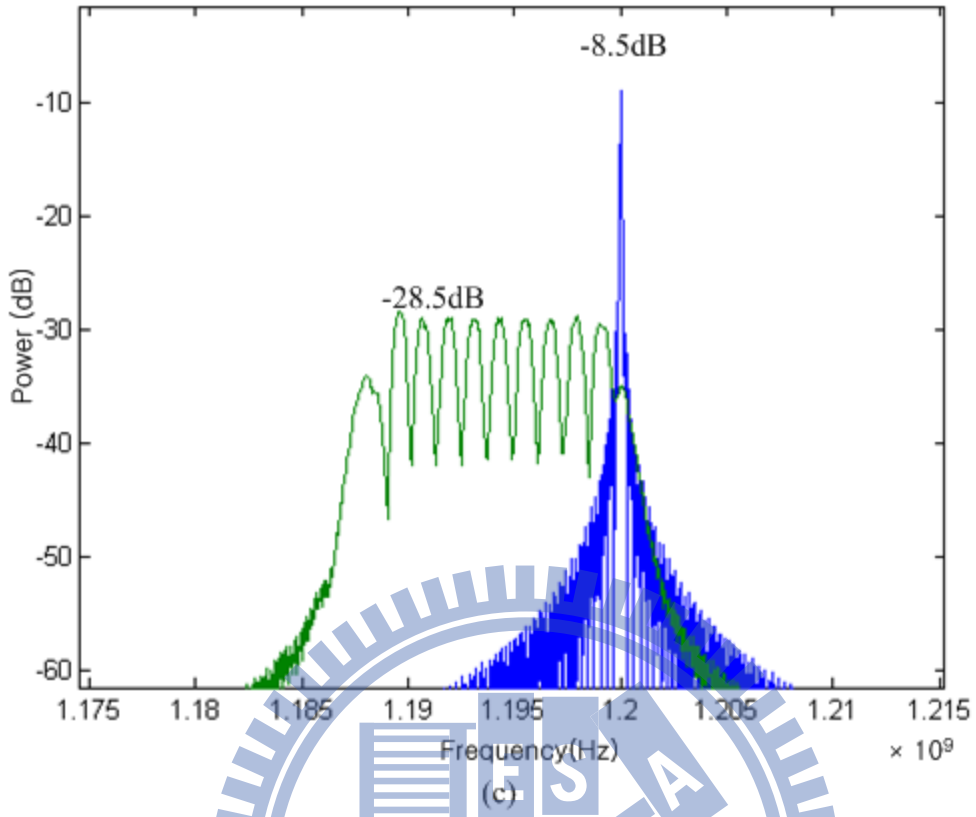


Fig. 3.28 Spectrum of PLL with down spread mode and 10000ppm frequency deviation (a) paper [12] with 40 steps (b) proposed method with 40 steps (c) paper [12] with 10 steps

Chap4

The Circuit Implementation of AD-SSCG

4.1 The Pseudo SSCG Square Wave

We design the modulated clock by two phase switching in our design. And the time to switch phase is when we sense the clock rising and falling edge. However, the conventional PFD compares the phase and frequency difference just at clock rising edge as shown in Fig. 4.1. Due to this feature, we can use pseudo modulated clock instead of the original modulated clock. The concept is shown in Fig. 4.2. The modulated clock is derived from CLK_REF by two phase switching. The delay time of one phase switching is T_{BUF}. Comparing to original modulated clock, the pseudo modulated clock only needs one phase switching. The delay time of one time phase switching is $2\,T_{BUF}$. We use the concept of pseudo modulated clock and the delay time of required delay BUF will be relaxed. Originally, the required delay BUF for 5000ppm frequency deviation is 25ps. Now, the delay BUF we need is only 50ps. In some specified process, it will solve the problem of the required minimum delay of BUF. Fig. 4.3 shows an example that is original modulated_CLK and pseudo modulated_CLK. The sequence is comprised of 1 modulated clock and 4 CLK_REF. We will find out that the PFD will regard these two modulated_CLK as the same one because the PFD only compares at the clock rising edge.

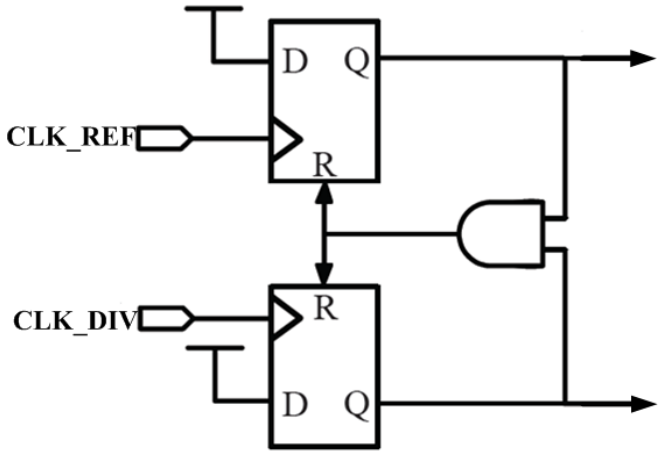


Fig. 4.1 Conventional PFD circuit

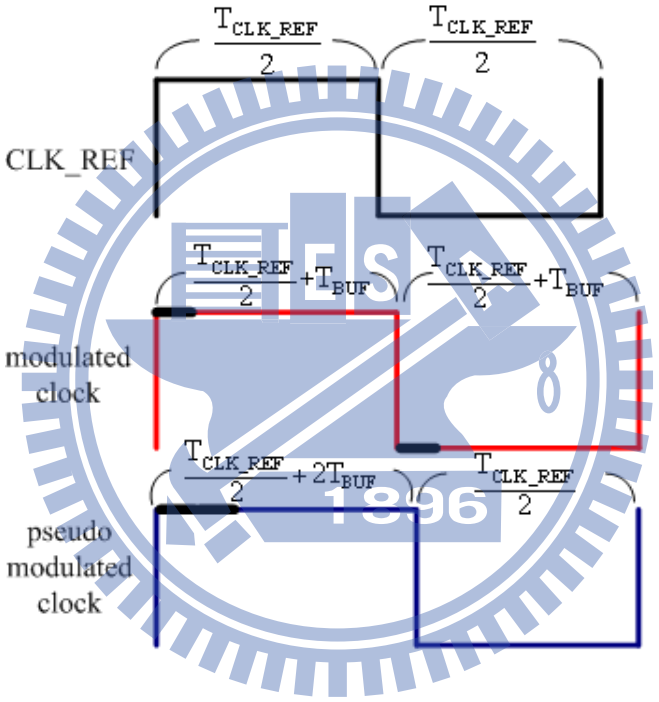


Fig. 4.2 Comparison of original modulated clock and pseudo modulated clock

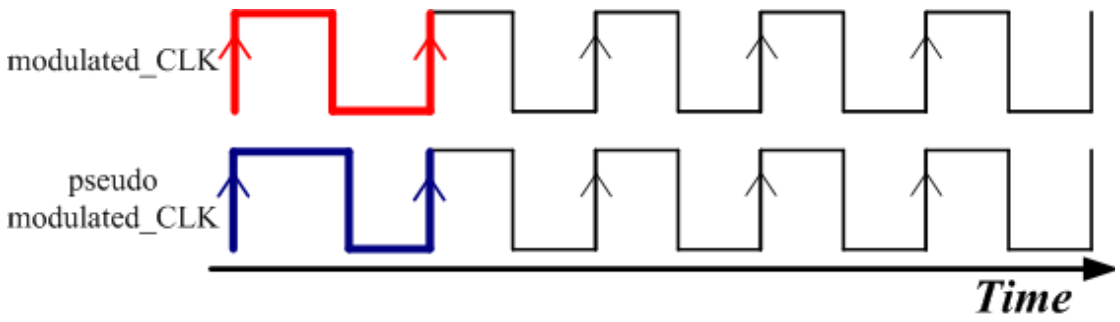


Fig. 4.3 Comparison of original modulated_CLK and pseudo modulated_CLK

Besides the advantage of relaxing the delay time of BUF, there is another advantage for control circuit of AD-SSCG. The control circuit is operated at double modulated_CLK which is 200MHz. If we use the concept of pseudo modulated_CLK the control circuit of AD-SSCG is just operated at 100MHz at clock rising edge. It can relax the timing of control circuit, and will gain advantage of area and power when we synthesize circuit.

4.2 The Algorithm of AD-SSCG

The algorithm of AD-SSCG is shown in Fig. 4.4. In the first block, "Phase" defines the amount of one phase switching, and the "Position" is the position of the DDLi. Our AD-SSCG can generate 5000ppm, 10000ppm, and 15000ppm. So when we set "Phase=1", it means AD-SSCG generate spread spectrum function with 5000ppm frequency deviation. We know the DDLi is like a multi-phase delay line with 230 phases (Positions). When "Position" is 100, it means that the 100-th phase is selected. So in the first step, you can decide the frequency deviation, and set the first phase of DDLi is 100 at the beginning.

Then, you can decide to turn on the "spread spectrum" function or not. If not, the "Position" will always be the same, and the AD-SSCG is just like a delay line. If we set "SSCG_ON=1", we still need to choose up or down spread spectrum mode. If the "Down_mode=1", it means down spread spectrum mode is selected, otherwise, up spread mode is selected.

A first–order ΣΔ modulator is used to randomly generate the position of modulated clock in a group. "Carry" is the output of first-order ΔΣ modulator. When the "Carry=1", we will switch to next phase, the amount of phase switching is decided by "Phase". When "Carry=0", the next "Position" is the same as the present "Position". If "Down_mode=1", the "Position" will be accumulated by a positive number at clock

rising edge. Upon accumulating "Position_max", AD-SSCG will find the other new clock rising edge on dummy DDLi and update the "Position". The "Position_max" and "Position_min" are respectively 227 and 3 because the maximum of switching phase is 3 when the frequency deviation of AD-SSCG is 15000ppm. The up spread spectrum mode is executed when "Down_mode=0", and the behavior is similar to "Down_mode". The only difference is that the "Position" is accumulated by a negative number.

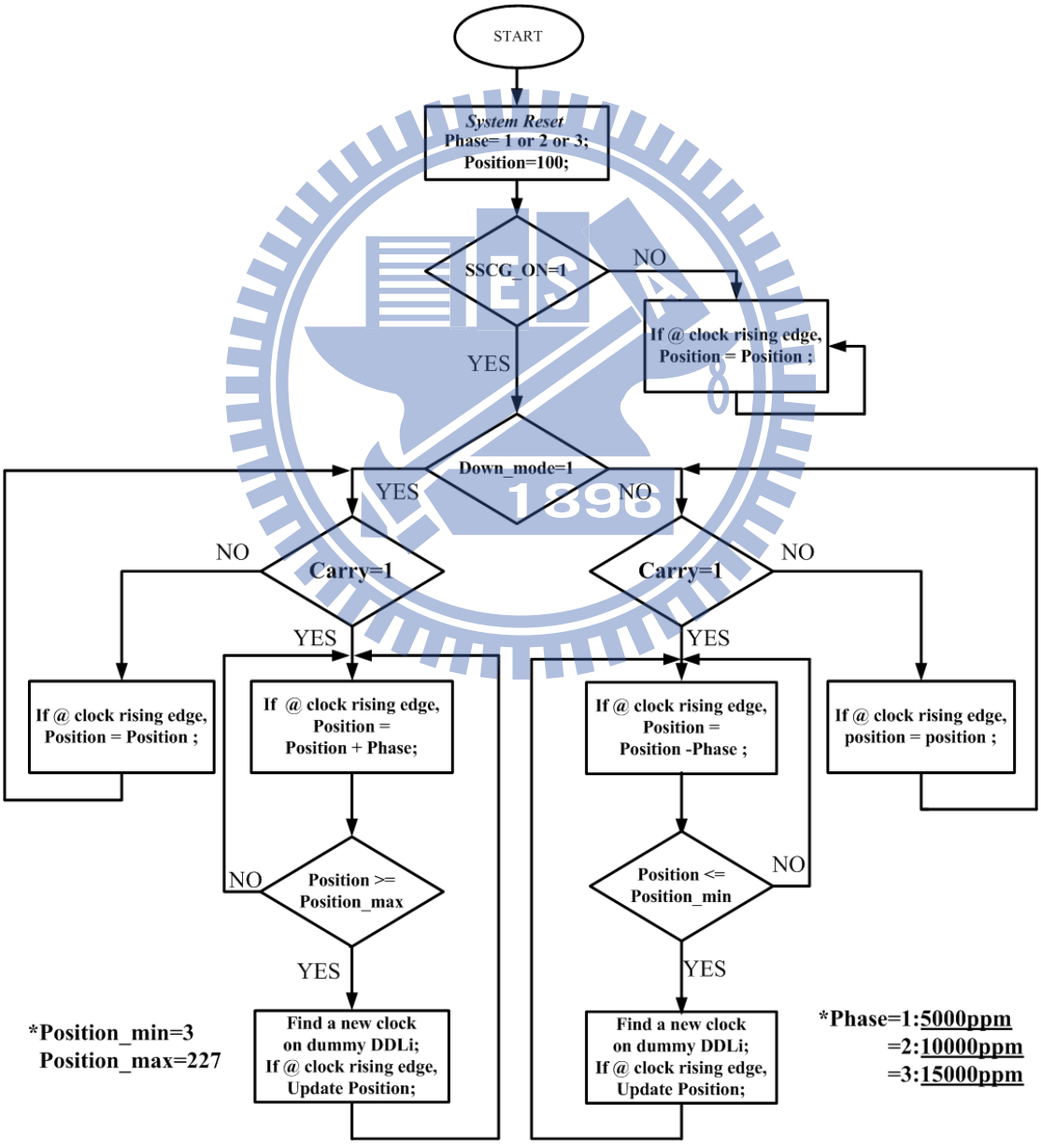


Fig. 4.4 The algorithm of AD-SSCG

4.3 Proposed Novel Coarse-Fine Delay Line Structure

In our AD-SSCG, we can generate a 100MHz modulated_CLK with 5000ppm frequency deviation. The period of 100MHz clock is 10000ps. Because we set the frequency deviation is 5000ppm, the delay time of the required T_{BUF} is 50ps (10000ps × 5000ppm = 50ps). In UMC90-CMOS process, the minimum delay time of T_{BUF} (two inverters) is around 40ps. We need 200 delay BUFs to construct the first part of DDLi and the second part of DDLi is 30 extra delay BUFs (The concept will be explained in section 4.4.) as shown in Fig. 4.5 (a).

We observe that this DDLi structure has 230 nodes. When 100MHz CLK_REF passes through the DDLi, a lot of nodes on DDLi will be charged and discharge. It will consume a lot of power. If we want to design a AD-SSCG with less frequency deviation, for example 2500ppm, the DDLi will have double nodes to complete this design. Besides, comparing to paper [12], it uses 27MHz CLK_REF, but we use 100MHz CLK_REF. Higher frequency CLK_REF will consume more power in DDLi structure. So we want to improve the power consumption of DDLi structure. Here, we call the DDLi structure in paper [12] as Uniform DDLi structure as shown in Fig. 4.5 (a). We present a new DDLi structure as shown in Fig. 4.5 (b) which names Coarse-Fine DDLi structure and has the same function. It can reduce the power consumption on DDLi.

Firstly, we need to design a C_BUF (Coarse_BUF). The delay time of C_BUF has a multiple factor of F_BUF (Fine_BUF). Here, we call original BUF in Uniform DDLi structure as F_BUF. In our design, the delay time of C_BUF is 1000ps which is 20 multiples of F_BUF. We can see in Fig. 4.5 (b), 20 F_BUFs are replaced by one C_BUF. The total delay time of Coarse-Fine DDLi structure is the same as Uniform DDLi structure. (10×C_BUF + 30×F_BUF = 10×(20×F_BUF) + 30×F_BUF= 230

F_BUF). Original 230-1 MUX is replaced by one 10-1 MUX (MUX-Coarse) and 30-1 MUX (MUX-Fine). In Uniform DDLi structure, the method of phase switching is one by one gradually. In Coarse-Fine DDLi structure, the operation mechanism is explained as follows. The modulated_CLK is at the output of MUX-Fine. Firstly, we fix the path of MUX-Coarse, and the phase selection on MUX-Fine is from phase 1 to phase 19. But the next switching phase is not phase 20, we use a C_BUF to replace 20 F_BUF. The procedure is that the phase selection of MUX-Fine go back to phase 1, and the phase selection of MUX-Coarse will switch to next phase. The 20-th phase to 30-th phase of MUX-Fine will only be used when the last phase of MUX-Coarse is selected.

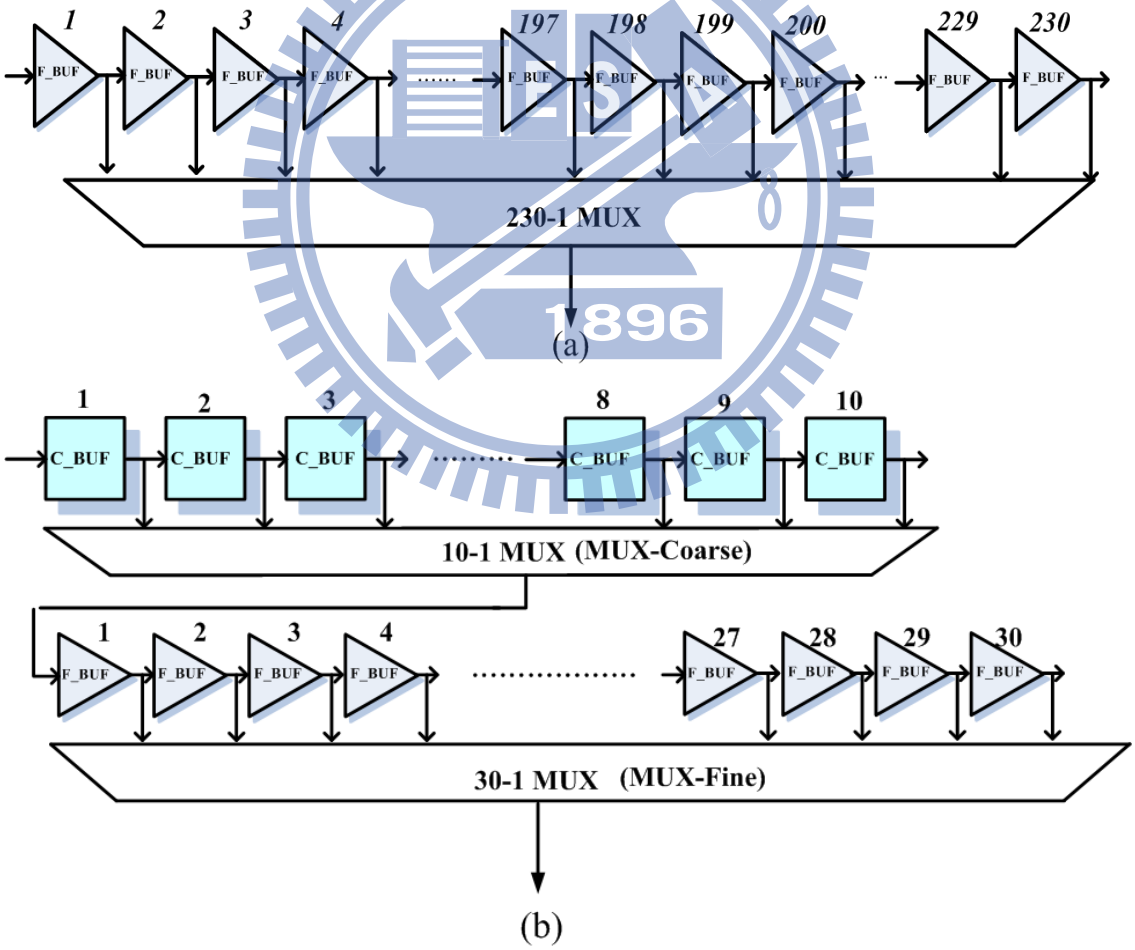


Fig. 4.5 (a) Uniform DDLi structure
(b) Coarse-Fine DDLi structure

Right now, we will provide some information to show the advantage of Coarse-Fine DDLi structure. We know the equation of power consumption of digital circuit is $\text{Power} = P_t \times \text{Freq} \times C_L \times \text{VDD}^2$. In this design, the parameters of CLK_REF frequency and VDD are fixed. The P_t defines the switching probability of the DDLi nodes. Because the input signal is a clock signal, P_t is also a fixed value. We will derive that the power consumption is only relative to C_L and the numbers of nodes of DDLi. In our implementation, the output capacitor of C_BUF 4 multiples of F_BUF. Here, we label Cap_F and Cap_C as the output capacitor of F_BUF and C_BUF, and Cap_C = 4×Cap_F. From Equation (4.1), we can improve power consumption by 3.3.

$$\frac{\text{Power}_{\text{Uniform}}}{\text{Power}_{\text{Coarse-Fine}}} = \frac{\#\text{ of nodes} \times C_L}{\#\text{ of nodes} \times C_L} = \frac{230 \times \text{Cap\_F}}{10 \times \text{Cap\_C} + 30 \times \text{Cap\_F}} = \frac{230 \times \text{Cap\_F}}{70 \times \text{Cap\_F}} \cong 3.3 \quad (4.1)$$

Besides saving power consumption, Coarse-Fine structure also saves chip area. The F_BUF uses two INVs, so the number of unit transistor is 8 (PMOS:NMOS = 3:1). The number of unit transistor of C_BUF is 24 (The circuit of C_BUF is shown in section 4.5 Fig. 4.15 (a)). The main circuit of C_BUF is 12 and extra PMOS and NMOS of C_BUF is also 12. From equation (4.2), we find the Coarse-Fine structure will improve chip area by 3.83.

$$\frac{\text{Area}_{\text{Uniform}}}{\text{Area}_{\text{Coarse-fine}}} = \frac{\#\text{ of unit transistor}}{\#\text{ of unit transistor}} = \frac{8 \times 230}{24 \times 10 + 8 \times 30} = \frac{1840}{480} \cong 3.83 \quad (4.2)$$

Based on analysis of equation (4.1) and (4.2), Coarse-Fine DDLi structure can achieve power and area saving at the same time.

4.4 The PVT Immunity Design Method

In this section, we discuss how to decide the length of dummy DDLi to cover PVT variation. There are two kinds of phase selection. One is down spread spectrum mode, and the phase selection is from left to right. The other one is up spread spectrum mode, and the phase selection is from right to left.

Firstly, we consider about down spread mode. In Fig. 4.6 (a), the vertical arrow presents clock rising edge. When there is no phase to switch, the clock rising edge will appear in the end of right hand side on DDLi. In ideal situation, the next clock rising edge will appear at point A. However, due to PVT variation, the delay of DDLi will be changed. The clock rising will appear at point B or C when the delay of DDLi is decreasing or increasing. But if the clock rising edge appears at point B, the clock rising edge is out of the DDLi. The edge detection circuit will not find out the clock rising edge. So, the required delay length of DDLi will make sure the circuit can be operated in PVT variation. The delay length of DDLi is designed for that we can tolerate +/- 15% delay variation as shown in Fig. 4.6 (b). It means that even the delay of DDLi is changed by +/- 15%, the function of AD-SSCG can still work.

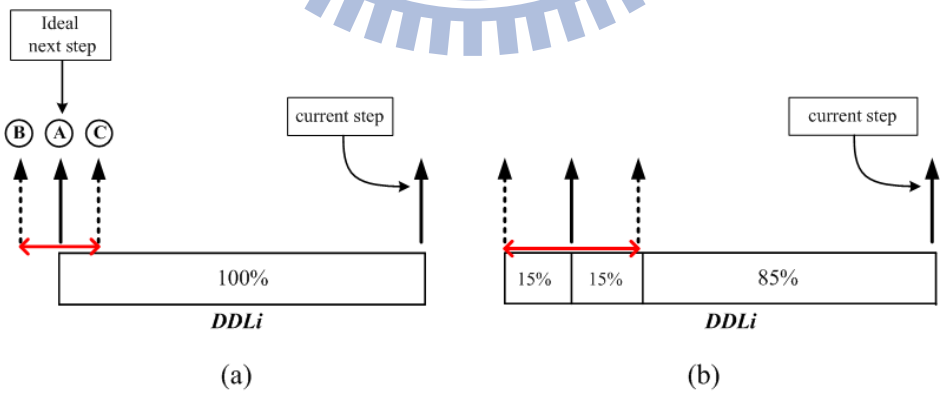


Fig. 4.6 (a) Clock rising edge on DDLi with PVT variation in down mode
(b) Clock rising edge on DDLi with +/- 15% PVT Variation in down mode

The dummy DDLi is used for detecting the next clock rising edge, and it should cover +/- 15% PVT variation. The dummy DDLi is 30% of the total DDLi in down

spread mode, as shown in Fig. 4.7. The connected DFFs are edge detection circuit.

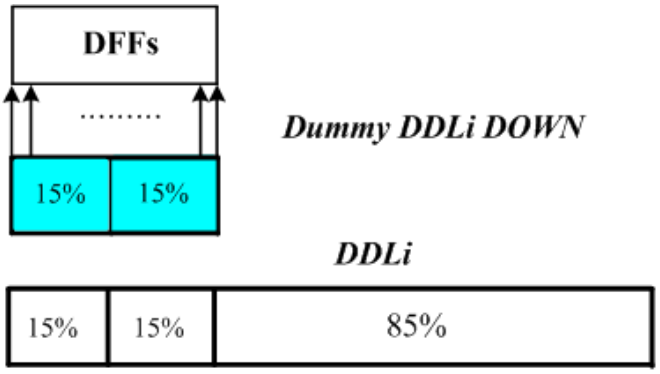


Fig. 4.7 DDLi and dummy DDLi in down mode

In up spread mode, the direction of phase switching is from right to left. When the last clock rising edge appears in the end of left hand side on DDLi, the next clock rising edge will appears at point A in ideal situation as shown in Fig. 4.8 (a). Due to PVT variation, the delay time of DDLi will be changed. The clock rising edge will appear at point B or C. Similarly, the delay length of DDLi is designed for that we can tolerate +/- 15% delay variation as shown in Fig. 4.8 (b).

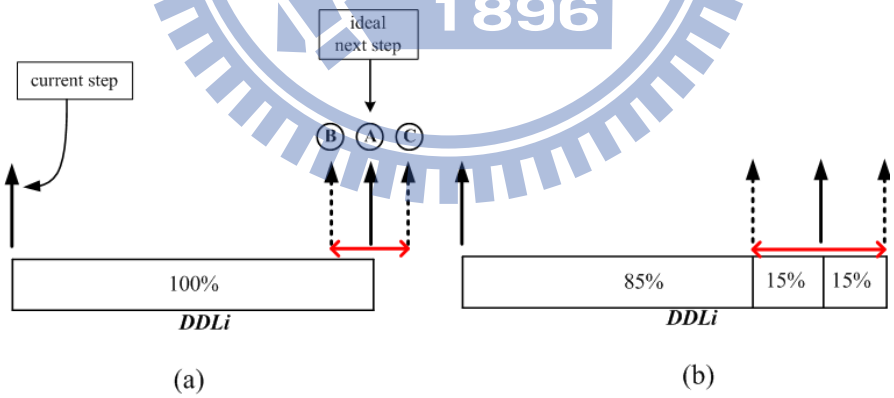


Fig. 4.8 (a) The clock rising edge on DDLi with PVT variation in up mode
(b) The clock rising edge on DDLi with +/- 15% PVT variation in up mode

We use the same concept to design dummy DDLi for up spread mode. The dummy DDLi is 115% of total DDLi as shown in Fig. 4.9. The end of right hand side 30% delay cell will be connected to edge detection circuit.

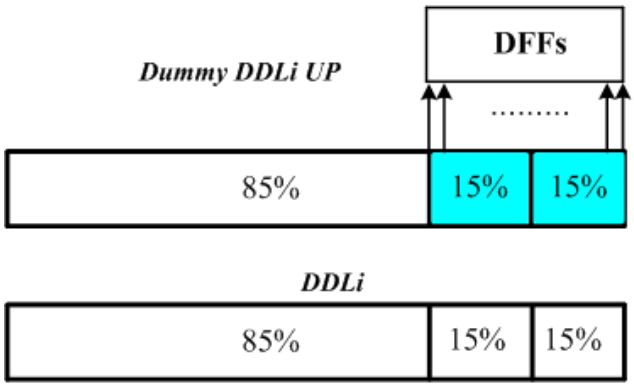


Fig. 4.9 DDLi and dummy DDLi in up mode

After discussing these two modes, we have decided the delay length of dummy DDLi. These two dummy DDLis will combine into one dummy DDLi which can be used for down and up spread modes. Fig. 4.10 is the overall architecture of DDLi and dummy DDLi.

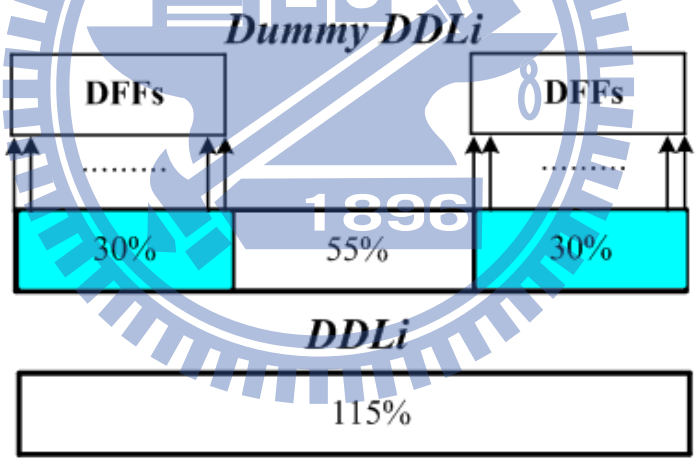


Fig. 4.10 Overall architecture of DDLi and dummy DDLi

The delay BUF of DDLi in AD-SSCG is designed by using simple digital circuit logic. We use the PVT immunity mechanism we described in this section to make sure that AD-SSCG will operated successfully with +/- 15% delay time variation. However, the spread spectrum amount will be changed if the delay time of BUF is changed. For example as shown in Fig. 4.5 (b), in ideal situation, the delay time of Fine BUF is 50ps so the period of modulated clock in maximum frequency deviation group of

modulated_CLK will be 10050ps. The spread spectrum amount is 5000ppm. However, if the delay time of Fine BUF is changed to 45ps, the period will be changed to 10045ps. The spread spectrum amount will be 4500ppm. The performance of spread spectrum will depend on the variation of delay time of delay BUF. Besides, due to the variation, the variation of delay time of 20 Fine BUFs and 1 Coarse BUF may not have consistency. It will generate one non-ideal modulated clock to effect spread spectrum performance.

4.5 Architecture of AD-SSCG

The overall architecture of AD-SSCG is shown in Fig. 4.11. The "Position Decision" block decides the turn on path of "MUX_Coarse" and MUX_Fine". The modulated_CLK is the end of DDLi which is treated as a clock for digital control circuit. If we use pulse-swallow mechanism to generate modulated_CLK, the generated modulated clocks in each group will be very regular. First-order ΔΣ modulator is used to randomize the regular modulated clocks, and it will push the spur noise from lower band to higher band. We can use accumulator to implement ΔΣ modulator, and the overflow bit of ΔΣ modulator is "Carry" in Fig. 4.11. When "Carry" is "1", the "Position Decision" will switch to next phase. Otherwise, when "Carry" is "0", the phase will be the same. The 100MHz CLK_REF passes through the DDLi and dummy DDLi and the DDLi consists of "Coarse DDLi" and "Fine DDLi". The dummy DDLi is used when there is no phase to switch. In normal case, that is, we have next phase to switch, the "Control" block will generate control signal "dummy_en=0" to turn off "dummy DDLi". It will disable the CLK_REF to pass through the dummy DDLi, and save power consumption. In case of no phase to switch, the "Control" block will generate control signal "dummy_en=1" to turn on the "dummy DDLi". When CLK_REF passes through dummy DDLi, we can find out that

there is the other clock on the dummy DDLi. We use DFFs to detect the position of the clock rising edge. The information of the position will be sent to "Position Encoder" block. In the next triggering clock, "Position Decision" will be updated when "circle_en=1".

In our design, some of the circuits are implemented by full custom design. For example, the accuracy of C_BUF and F_BUF will decide the final SSCG performance. So, the "DDLi", "Dummy DDLi" and "DFFs" will be designed by full custom design. The control parts of AD-SSCG are implemented by standard cell design flow.

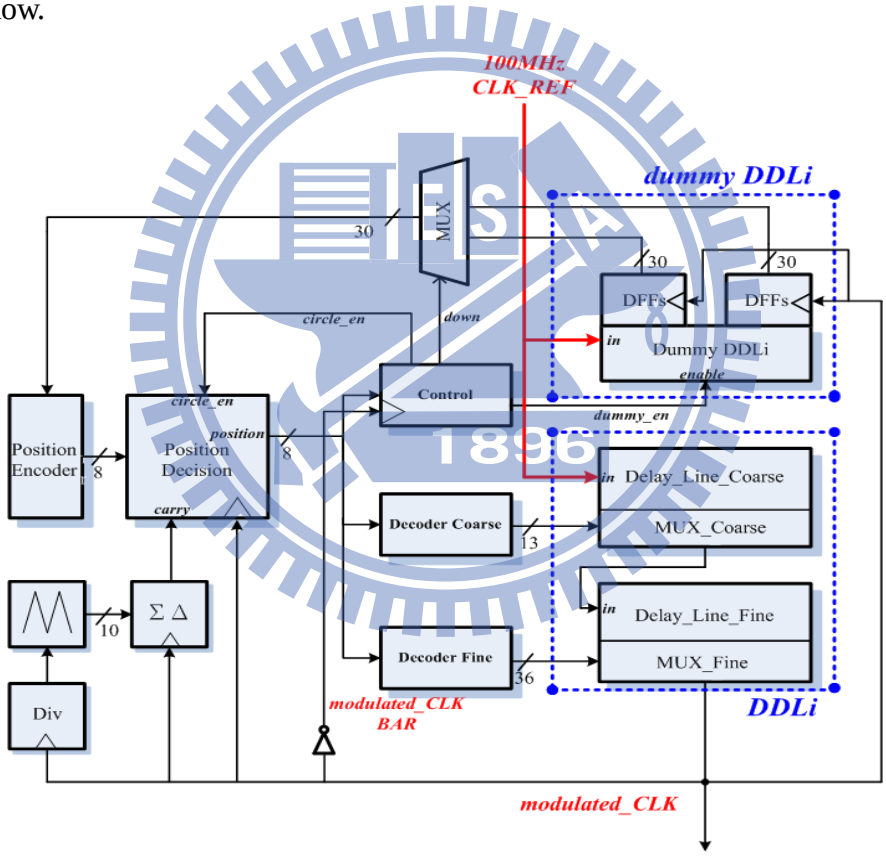


Fig. 4.11 The overall architecture of AD-SSCG

4.5.1 Control Implementation

The circuit of control part is shown in Fig. 4.12. In this section, we will show circuit simulation. Case 1 is the case of normal phase switching. Case 2 is the case when there is no next phase to switch.

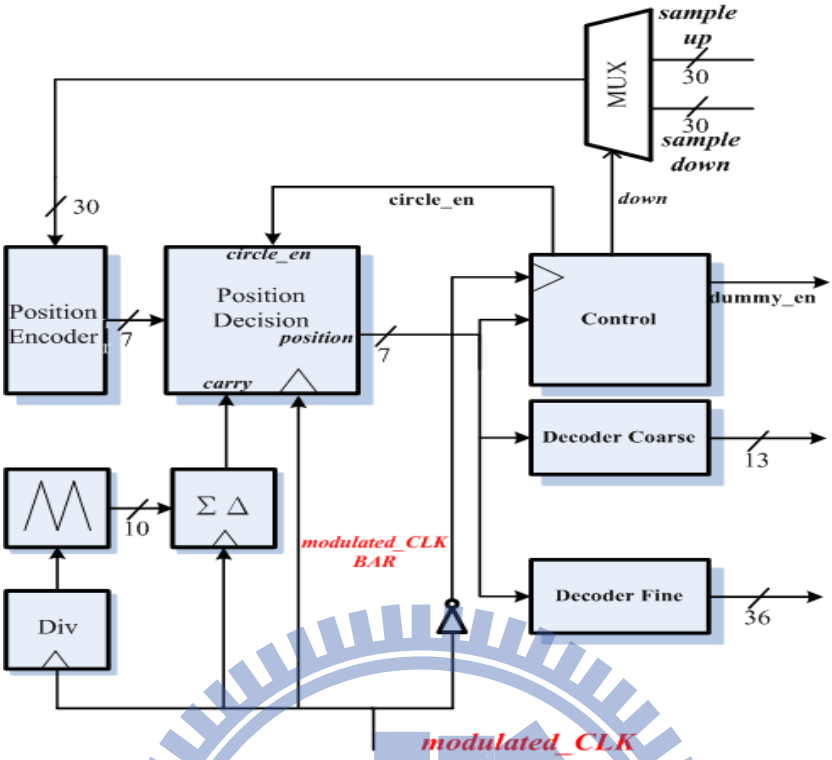


Fig. 4.12 The circuit of control

We use CAD tool Verdi to show the pre-sim results. We set the phase=1 which means the frequency deviation is 5000ppm, and set the down spread spectrum mode. The simulation of case 1 is shown in Fig. 4.13. When the modulated CLK triggers the control circuit and the Carry signal is also "1", the position [7:0] signal will switch to next phase.

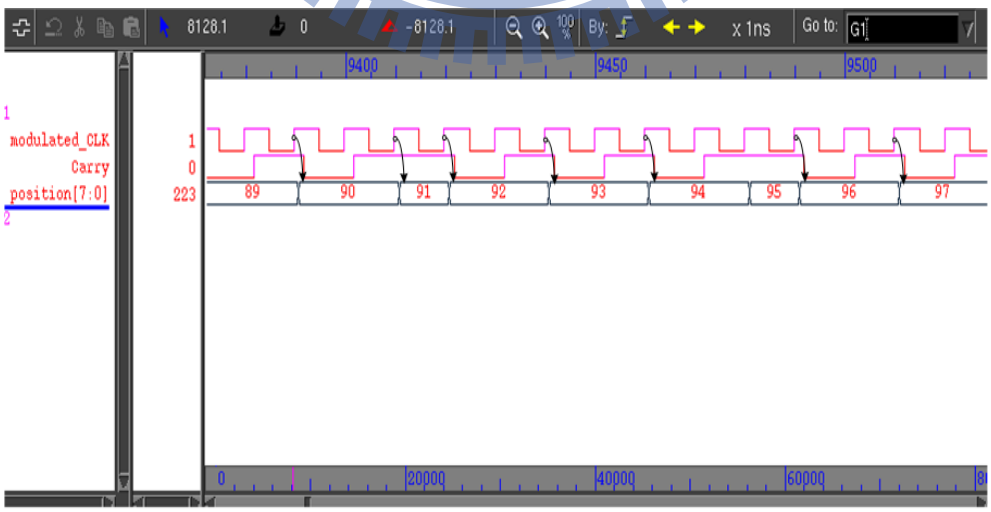


Fig. 4.13 Simulation of control circuit of case1 (pre-sim)

The simulation of case 2 is shown in Fig. 4.14. The position [7:0] is 226 which is

the last phase to be selected on DDLi. The sample [7:0] is 25 which is the position of next clock rising edge, and will be sent to “Position Decision” at next triggering modulated_CLK. We can observe that when “circle_ENABLE=1” the position [7:0] will be updated to 26 (25+1). The new position [7:0] shall be 26 instead of 25 because we have to include the switching phase when “Carry=1”.

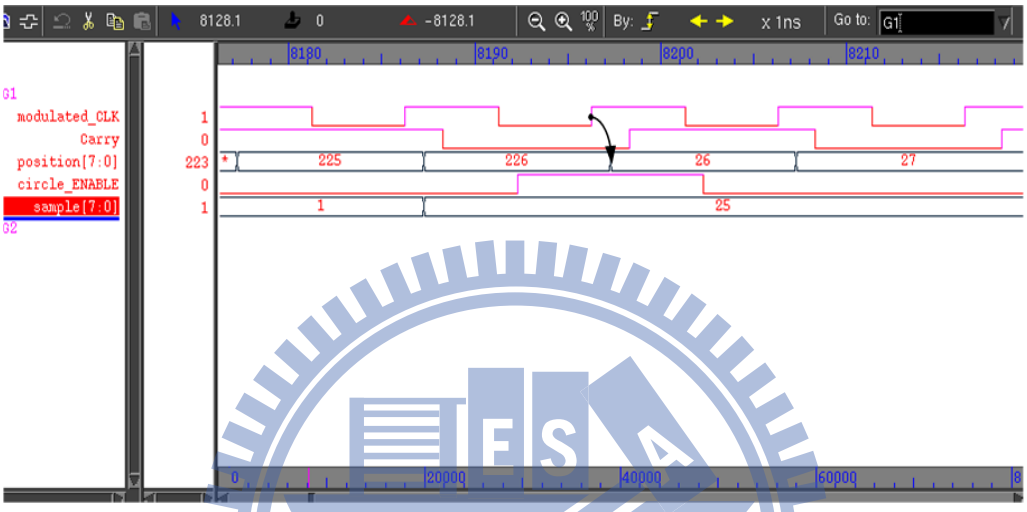


Fig. 4.14 Simulation of control circuit of case 2 (pre-sim)

4.5.2 DDLi and Dummy DDLi Implementation

This is the DDLi and dummy DDLi part of AD-SSCG as shown in Fig. 4.15. The two groups of DFFs in dummy DDLi are used respectively for down and up spread modes. The Q output of DFFs will save the information of next clock rising edge. Then, it will be sent to “Position Encoder“ block in control circuit. There is a TRI_INV in front of dummy DDLi. When “dummy_en=1” the TRI_INV will be turn on, the CLK_REF will pass through dummy DDLi. The 10-1 MUX and 30-1 MUX is constructed by TRI_INVs. This structure of MUX will reduce the delay time and area. In our design procedure, we also consider about the loading of every node in MUX to make sure the driving ability is enough.

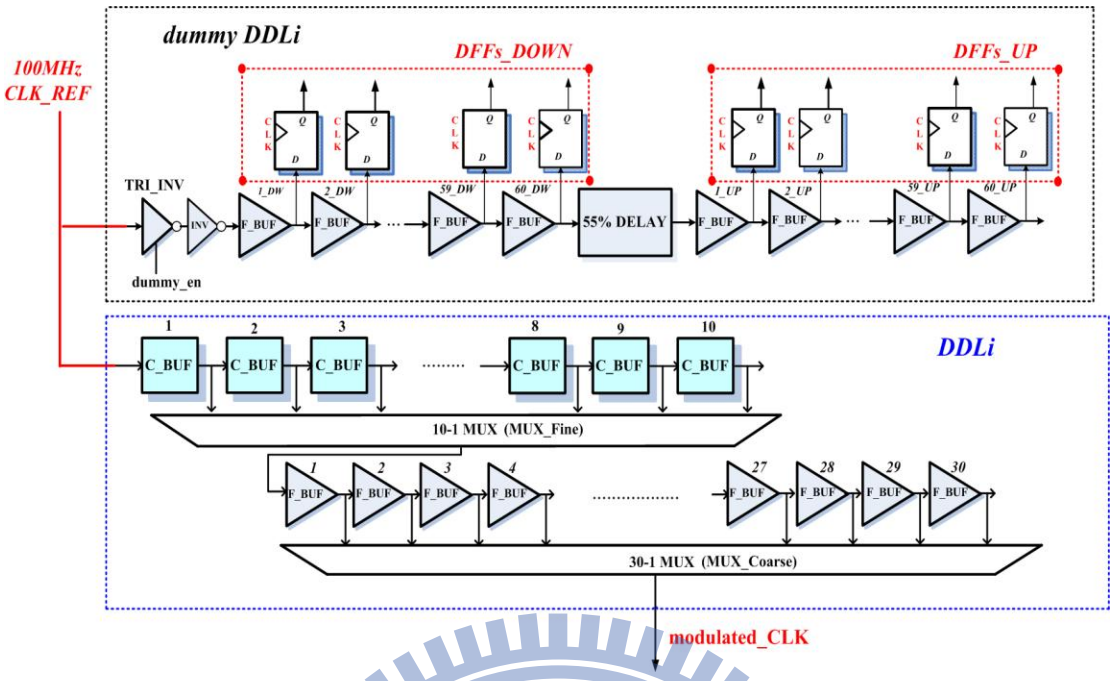


Fig. 4.15 The circuit of DDLi and dummy DDLi

The F_BUF is designed by two INVs. The delay time of F_BUF is 50ps in post-sim condition. The C_BUF references the delay cell of paper [14] which is a low power Schmitt trigger circuit. The delay time of Schmitt trigger circuit is around hundreds ps. In order to meet 1000ps, we connect NMOS and PMOS transistor to the output of Schmitt trigger circuit as extra capacitor to complete C_BUF as shown in Fig. 4.16 (a). The circuit of TRI-INV we use in MUX and dummy DDLi is a C^2_MOS circuit as shown in Fig. 4.16 (b).

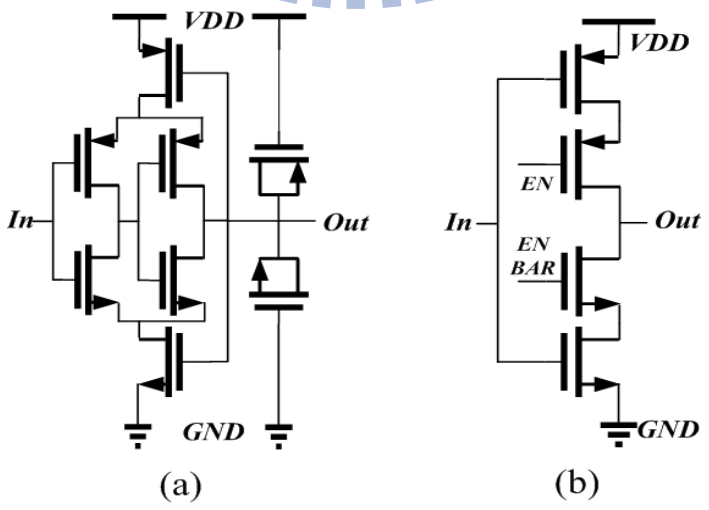


Fig. 4.16 (a) The circuit of C_BUF (b) The circuit of TRI_INV

Fig. 4.17 and Fig. 4.18 show the delay time of C_BUF and F_BUF. In our design, the C_BUF and F_BUF are 1000ps and 50ps respectively in post-sim condition.

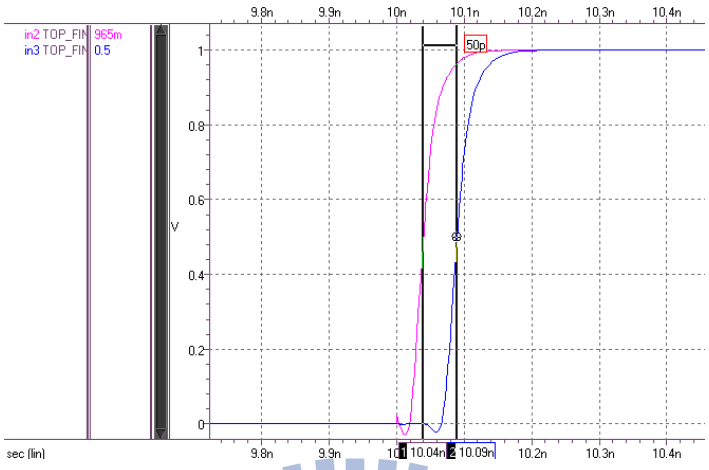
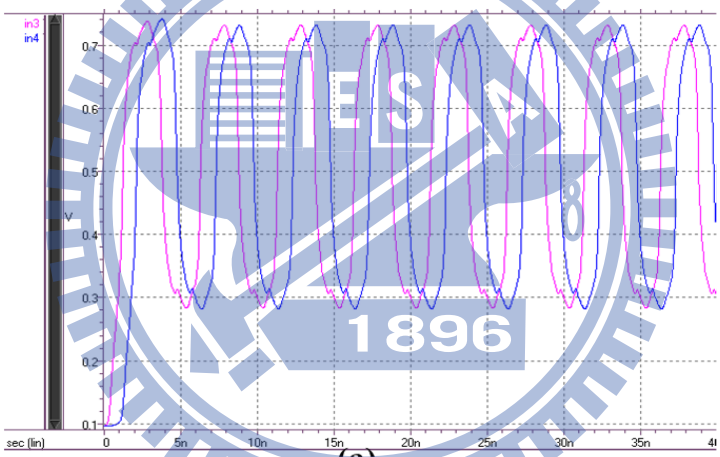
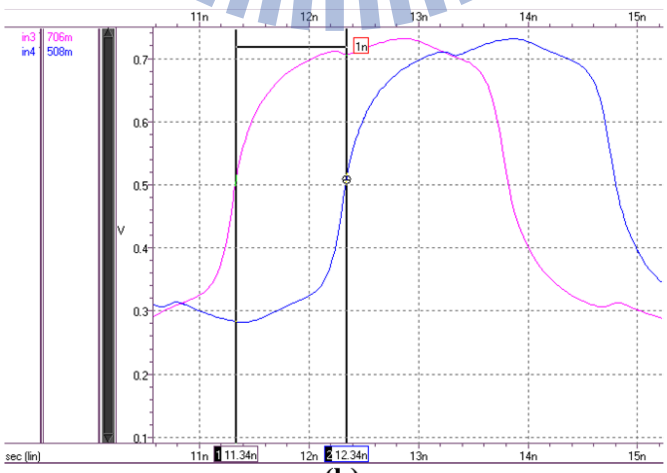


Fig. 4.17 The delay dime of F_BUF (post_sim)



(a)



(b)

Fig. 4.18 (a) The delay waveform of C_BUF (post_sim)
(b) The delay time of C_BUF (post_sim)

4.6 The Glitch Issue

When we switch phase, the incorrect switching time may cause glitch. Thus, we need to consider the correct switching time to avoid the glitch. In our design, we have two kinds of phase switching. Case 1 is from "leading phase" to "lagging phase" as shown in Fig. 4.19 (a), and Case 2 is from "lagging phase" to "leading phase" as shown in Fig. 4.19 (b). Here, ΔT is the time difference of two phases. The safe zone of phase switching of case 1 is between ΔT to $T_{CLK\_REF}/2-\Delta T$. T_{CLK_REF} is the period of CLK_REF. The safe zone of phase switching of case 2 is between ΔT to $T_{CLK\_REF}-\Delta T$. These two cases will both be included in our design, so we AND these two safe zones. The effective safe zone for our AD-SSCG is between ΔT to $T_{CLK\_REF}/2-\Delta T$.

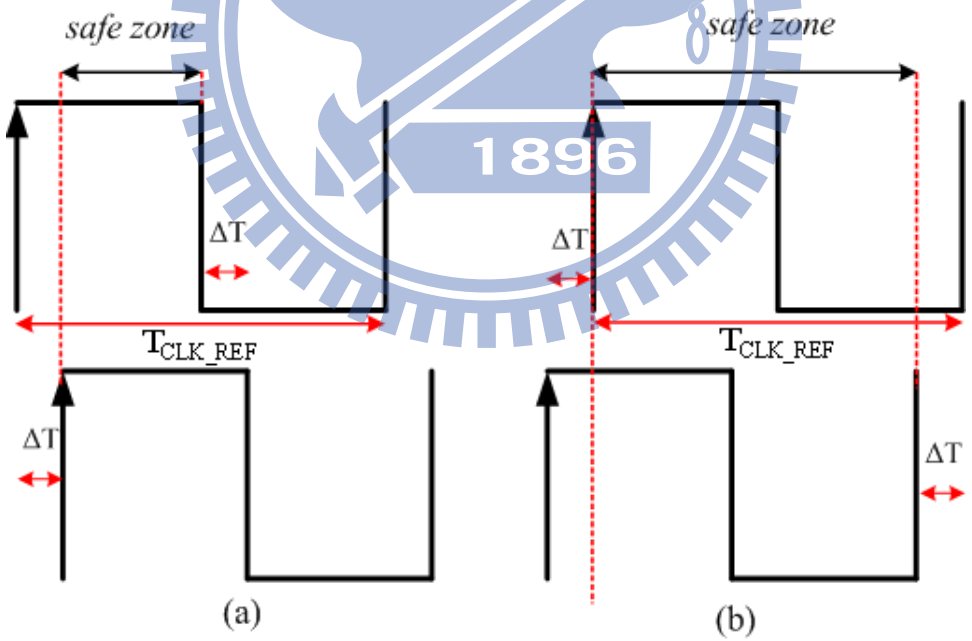


Fig. 4.19 (a) Phase switching from "leading phase" to "lagging phase"

(b) Phase switching from "lagging phase" to "leading phase"

Because we adopt the Coarse-Fine delay cell structure, the phase switching will occur between (1) F_BUF to F_BUF (2) C_BUF to C_BUF. Because the safe zone

between C_BUF to C_BUF is severer than F_BUF to F_BUF, we just consider the condition of phase switching of C_BUF. The delay time of C_BUF is 1000ps, and the period of CLK_REF is 10000ps. The safe zone of case 1 is between 1000ps to 4000ps, and the safe zone of case 2 is between 1000ps to 9000ps. We AND these two safe zones, the combined safe zone is between 1000ps to 4000ps. When we synthesize the control circuit of AD-SSCG with safe zone consideration, the control circuit should complete their function between 1000ps to 4000ps, even though the modulated_CLK=100MHz.

In critical path of control circuit, the first "Position Decision" block is triggered by modulated_CLK, then, the output signal of "Position Decision" connects to "Decoder Fine" and "Decoder Coarse" blocks. The total delay time is decided by (A)"Position Decision" and "Decoder Fine" and (B) "Position Decision" and "Decoder Coarse" as shown in Fig. 4.20. When we use Design Complier (CAD tool) to synthesize these blocks, we need to set design constraint for these three blocks. In our practical implementation, the "Position Decision" is synthesized by 1.3ns (1300ps) and "Decoder Fine" and "Decoder Coarse" are synthesized by 1ns (1000ps). The 2300ps delay time will pass the requirement of safe zone.

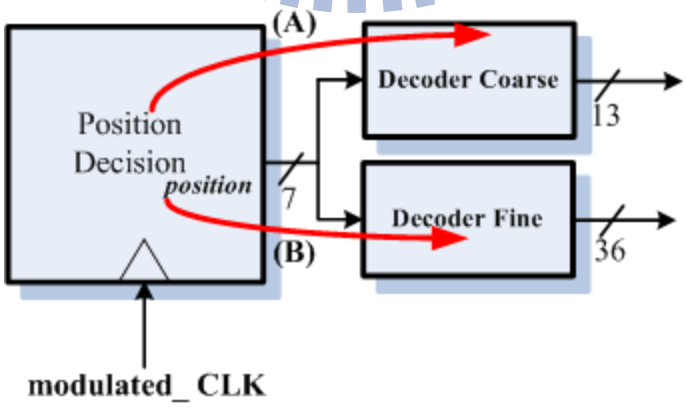


Fig. 4.20 The delay path to from "Position Decision" to two "Decoder"

Another kind of glitch in the design occurs due to practical synthesized circuit. As shown in Fig. 4.21, the 8 bits of position [7:0] of "Position Decision" don't arrive at

the same time. It will let C_X1 [10:0], C_X2 [1:0], F_X1 [29:0], and F_X2 [5:0] (the output signals of "Decoder Coarse" and "Decoder Fine") change the turn on path of MUX before settling the final correct path. The unstable value will turn on incorrect phases of DDLi to occur glitch.

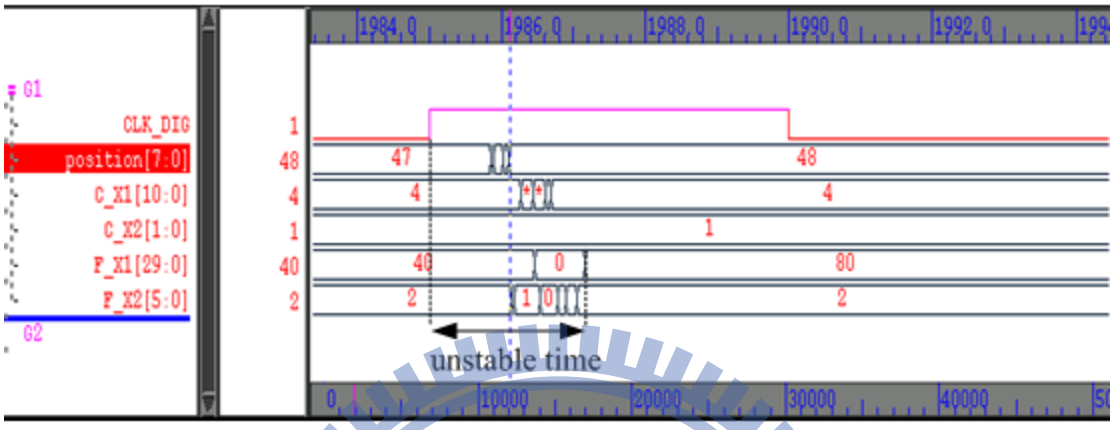


Fig. 4.21 The unstable value output of "Position Decision"

To solve this problem, if we know the required stable time of "Position Decision" and two "Decoder", we can sample the final stable value. We can use extra DFFs whose CLK is generated by modulated_CLK with a specified delay time to do this job as shown Fig. 4.22. If we know the signals need 2300ps to be stable, we can use the clock which is delayed with 3000ps from modulated_CLK to sample the final stable value. Thus, the glitch due to the signals don't arrive at the same time will not happen. Fig. 4.23 shows the revised architecture AD-SSCG.

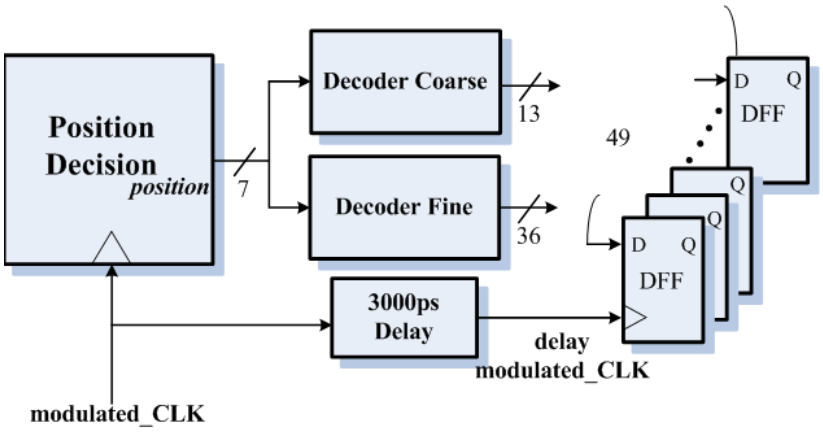


Fig. 4.22 The solution for glitch of unstable output value

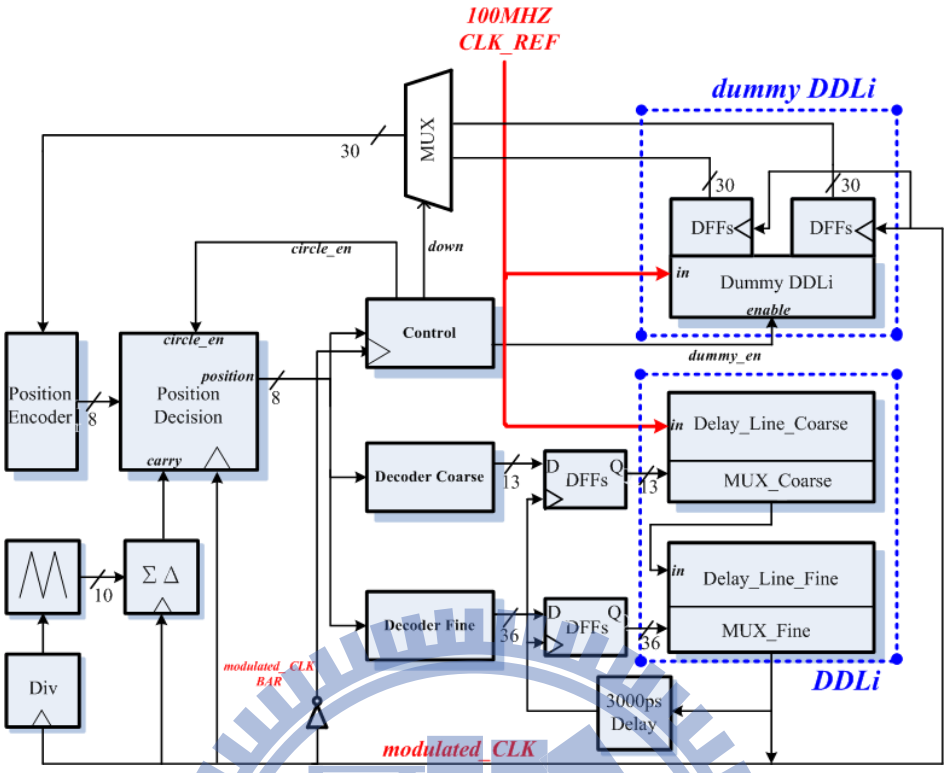


Fig. 4.23 The architecture of revised AD-SSCG

4.7 Experimental Results

4.7.1 Circuit Simulation

The AD-SSCG block is implemented with a 3GHz PLL [15]. The 3GHz PLL has already taped out and the function is also worked. The AD-SSCG will generate a 100MHz modulated_CLK, the mode is setting with down spread spectrum and 5000ppm frequency deviation. This AD-SSCG can be treated as an IP, it will let 3GHz PLL also have down spread spectrum function with 5000ppm frequency deviation. Table 4.1 shows the parameters of this 3GHz PLL. In order to verify the spread spectrum function, we firstly use simulink model of MATLAB to do simulation. Fig. 4.24 shows the locking condition of 3GHz PLL. Fig. 4.25 shows the spread spectrum profile of 3GHz PLL and relative spectrum plot. The EMI reduction of spreaded

3GHz clock is 21.5dB.

Table 4.1 The parameters of 3GHz PLL [15]

K_{VCO}	1000MHz/V
I_p	230 uA
C_1	108pf
C_2	8.3pf
R_1	2.3kΩ
N(divider ratio)	30
Input Frequency	100MHz
Output Frequency	3GHz

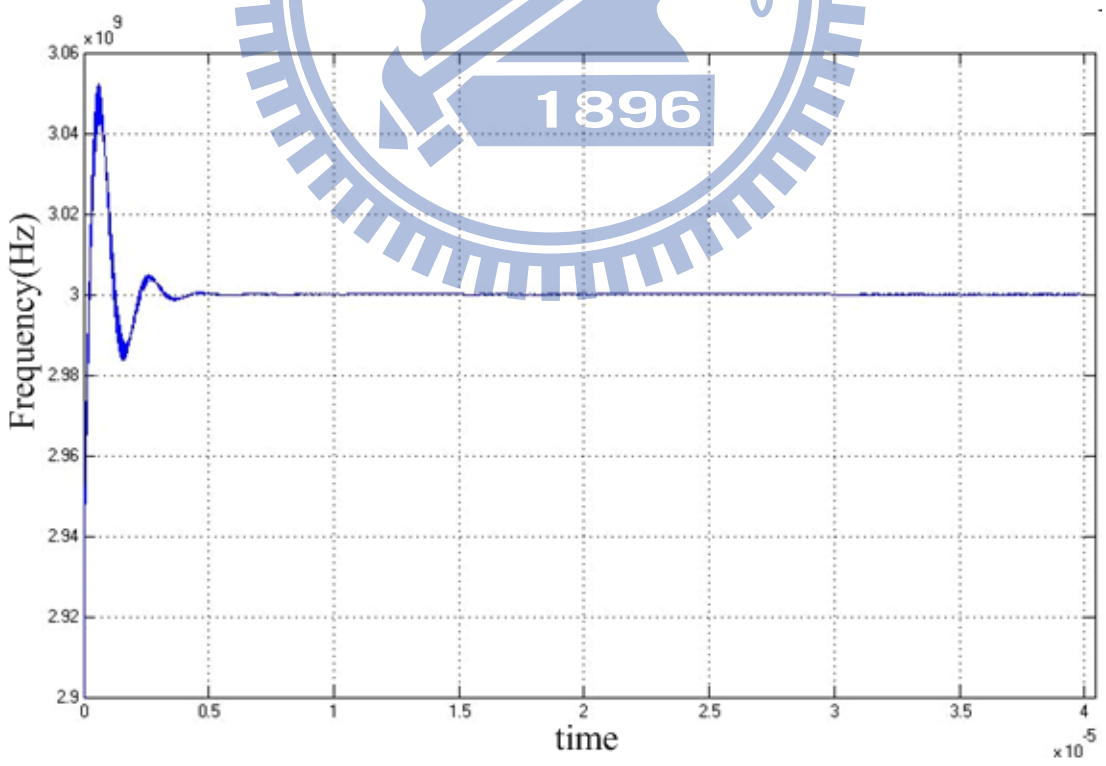


Fig. 4.24 Lock-in behavior of 3GHz PLL

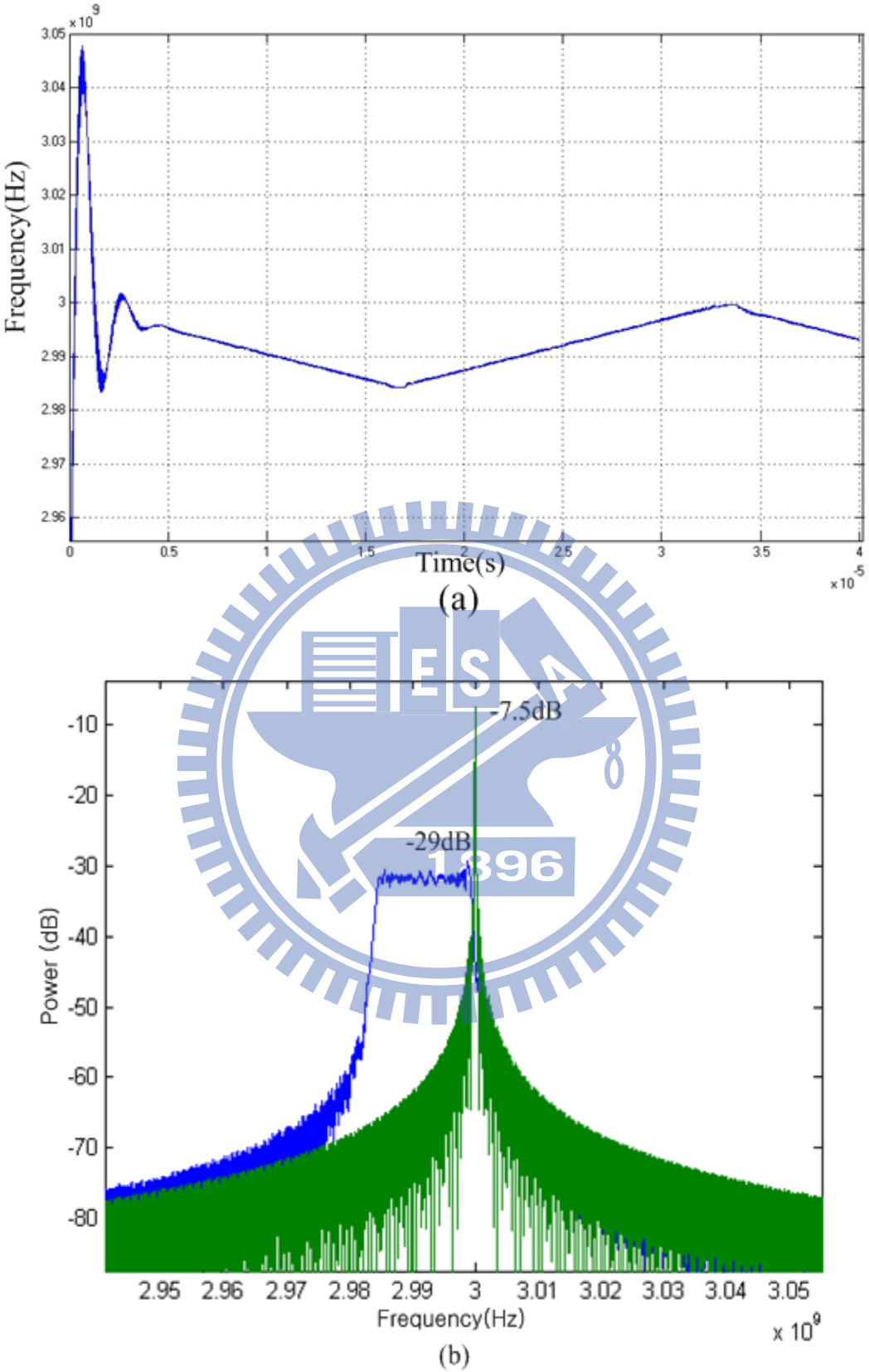


Fig. 4.25 (a) Frequency of 3GHz PLL with 5000ppm and down mode

(b) Spectrum of 3GHz PLL with 5000ppm and down mode

The post layout circuit simulation is carried out by “hspice”. Fig. 4.26 shows the Vc lock-in behavior with initial conidial Vc=0.5. Fig. 4.27 shows the output of 3GHz PLL, and the simulation results including the ground bounce. Fig. 4.28 shows the output of 3GHz BUF. Because the output of 3GHz BUF is connected to a “Bias Tee” device, the swing of signal is above VDD. Fig. 4.29 shows the peak-to-peak jitter of 3GHz PLL without spreading spectrum. Fig. 4.30 – 4.31 show the 3GHz PLL simulate in spread spectrum mode, and the setting is down spread spectrum with 5000ppm frequency deviation. Fig. 4.30 shows the behavior of Vc node, the profile is a triangular. Fig. 4.31 shows the peak-to-peak jitter of 3GHz PLL with spread spectrum. We observe the cycle-to-cycle jitter of spread 3GHz clock in some period, and the maximum and minimum are respectively 4.14ps and -3.94ps. Comparing to the original peak-to-peak jitter, the order of jitter is not increased by our AD-SSCG. Fig. 4.32 shows the spectrum of 3GHz PLL without and with spread spectrum, and the peak of spectrum respectively are -4dB and -26dB. The EMI reduction is 22dB. Fig. 4.33 the spectrum of 100MHz square waveform with and without spread spectrum. The EMI reduction is 8dB.

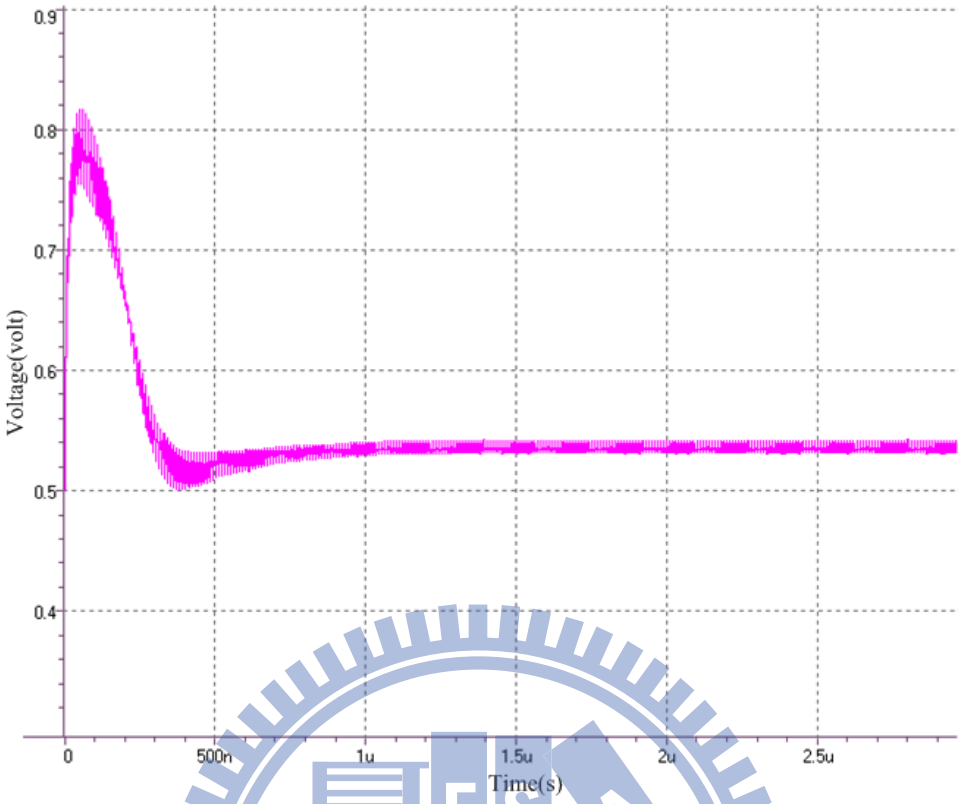


Fig. 4.26 Vc lock-in simulation of 3GHz PLL (post-sim)

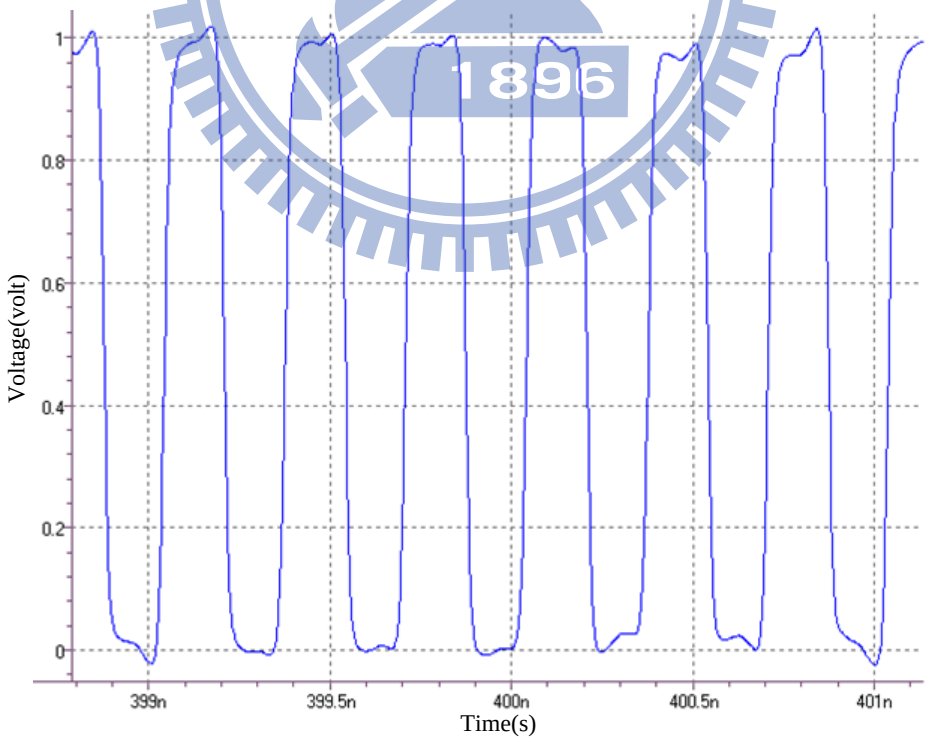


Fig. 4.27 The full-swing output of 3GHz PLL (post-sim)

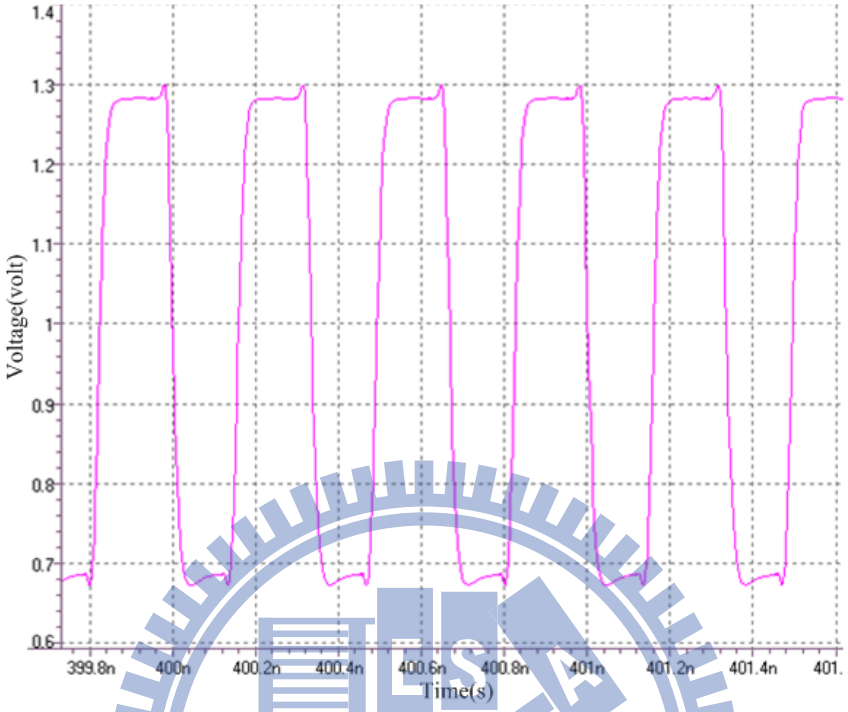


Fig. 4.28 The output of 3GHz BUF (post-sim)

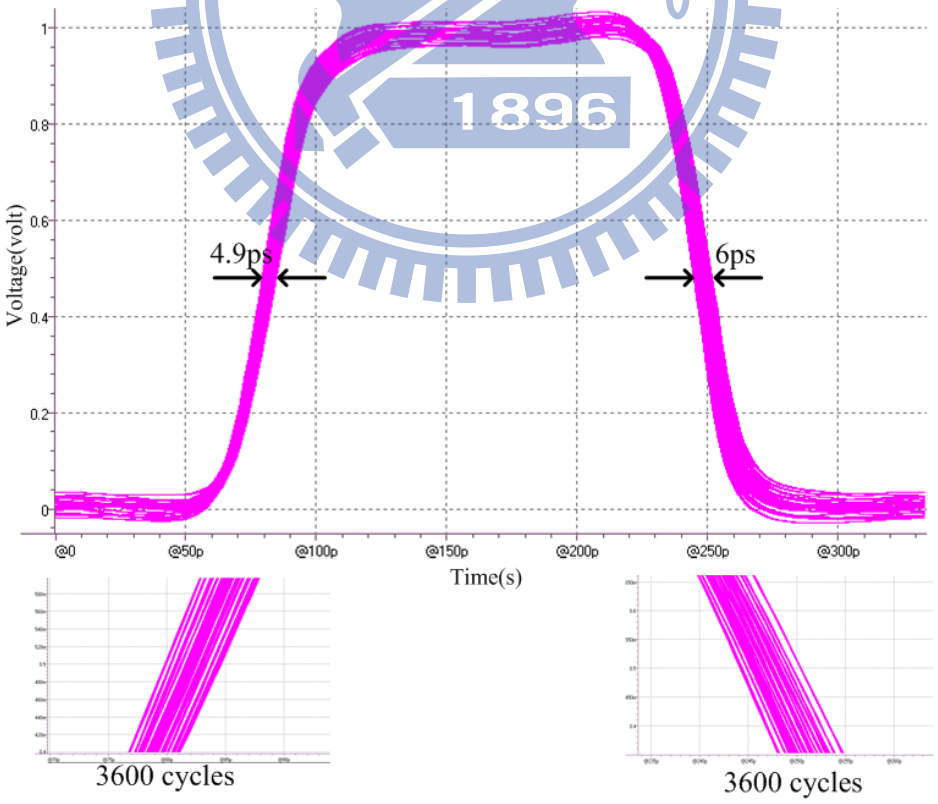


Fig. 4.29 Peak-to- peak jitter of 3GHz PLL (post-sim)

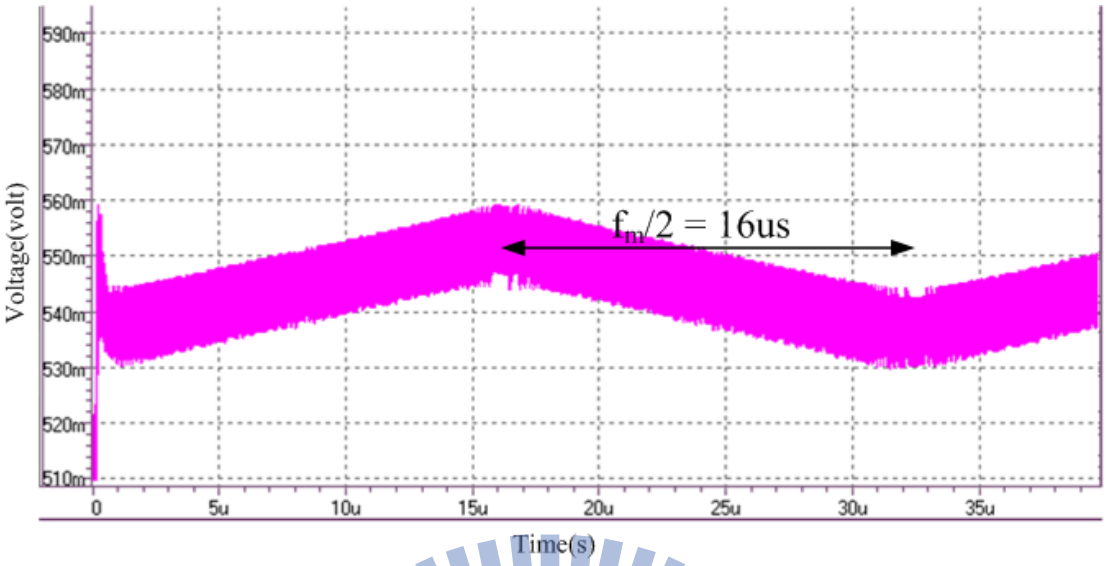


Fig. 4.30 Vc of 3GHz PLL with triangular profile (post-sim)

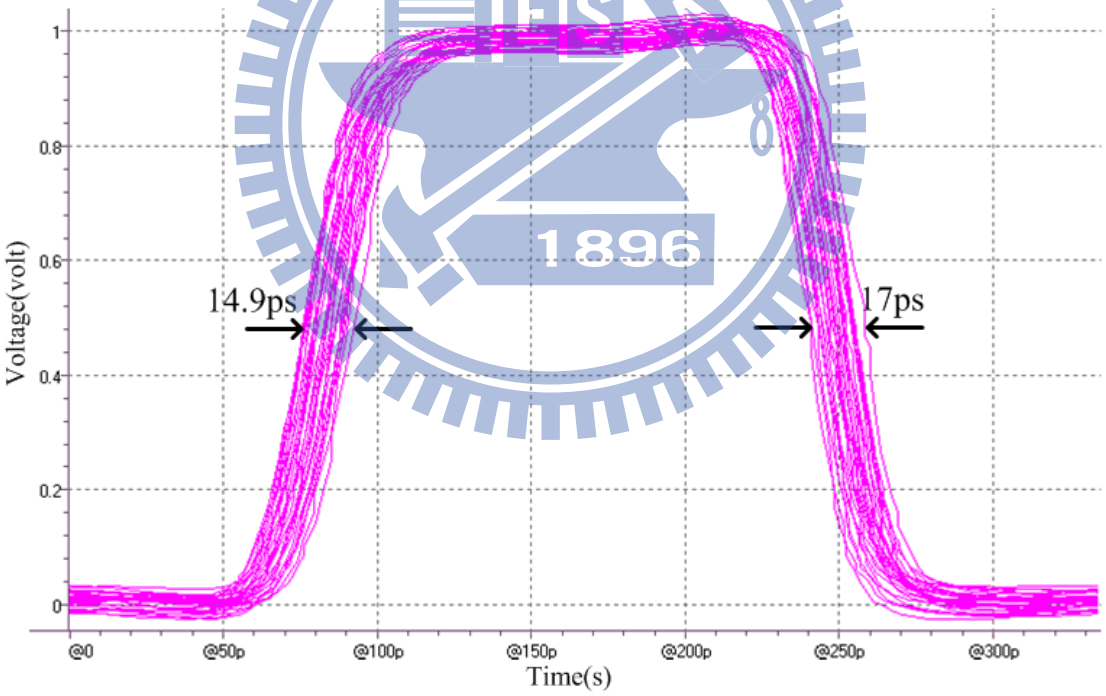


Fig. 4.31 Peak-to-peak jitter of 3GHz PLL with spread spectrum (post-sim)

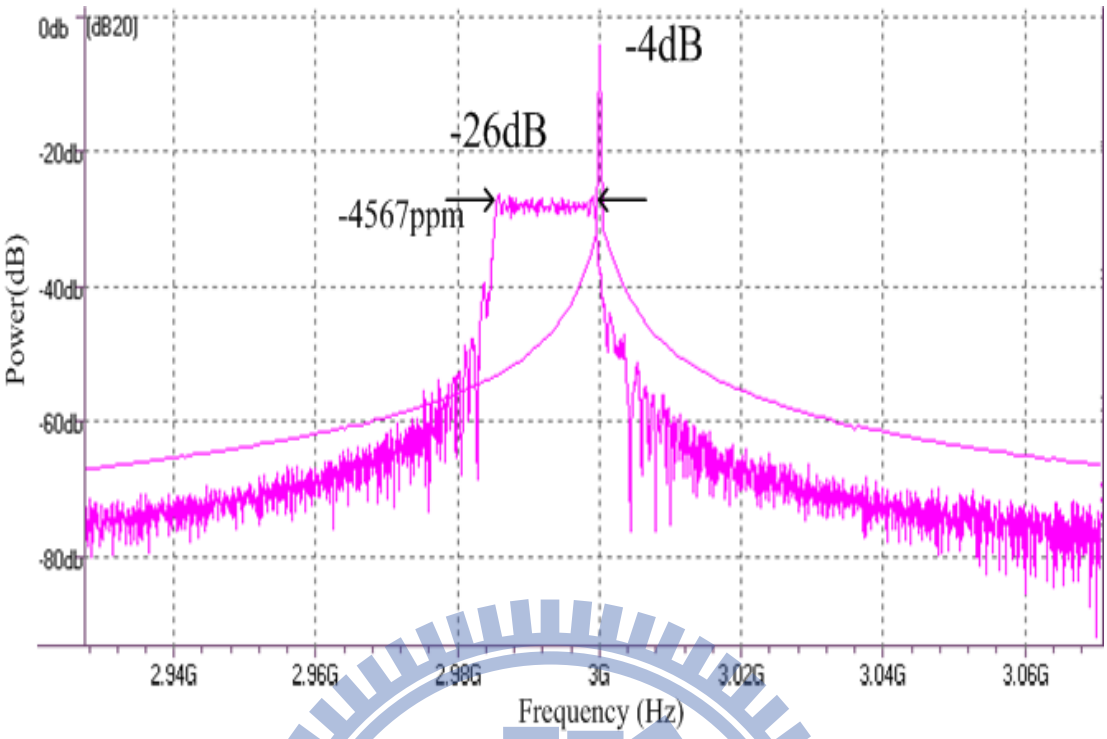


Fig. 4.32 The spectrum of 3GHz PLL with and without spread spectrum (post-sim)

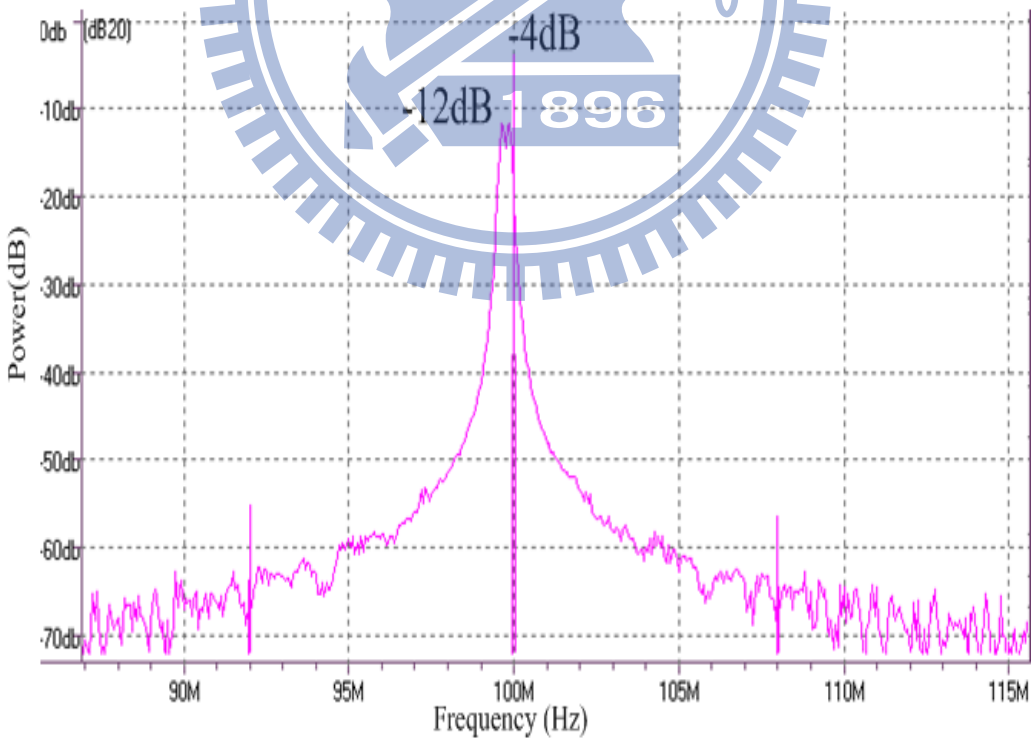


Fig. 4.33 Spectrum of 100MHz AD-SSCG square waveform
and 100MHz square waveform (post-sim)

The performance of total AD-SSCG and 3GHz PLL is shown in Table 4.2. The feature of proposed AD-SSCG show in Table 4.3.

Table 4.2 Design summary of Domino AD-SSCG and 3GHz PLL

Process	90nm
Supply Voltage	1V
Input clock	100MHz
Output clock	3GHz
Power	(1) PLL = 19.4mW PLL BUF = 35.1mW (2)AD-SSCG = 2.9mW
Area	Total: 950um × 705um(w/i pad) AD-SSCG: 335um × 105um
EMI Reduction	21.5dB (MATLAB)
Modulation Profile	Triangular
Modulation Type	Down mode
Modulation Frequency	31.5KHz
Frequency Deviation	5000ppm

Table 4.3 Design summary of Domino AD-SSCG

modulated_CLK	100MHz
Modulation Profile	Triangular
Modulation Type	Up/Down
Frequency Deviation	5000/10000/15000ppm
Area	Toatl: 335um × 105um Full Custom: 335um × 50um Digital Control: 310um × 55um (Gate Count = 2744)
Power	2.9mW@1V

4.7.2 Layout

The AD-SSCG is implemented with a 3GHz PLL in UMC CMOS 90nm 1P9M process. The chip area is 950um x 705um (w/I pad area). The overall layout is comprised of three parts as shown in Fig. 4.34. One is our AD-SSCG as shown in Fig. 4.35. We want to minimize the area of AD-SSCG so the layout style of full custom part is standard cell layout style. The digital control part is placed and routed by CAD tool "Soc_Encounter". The shape is set to match the full custom part. The location of 3GHz PLL is in the left region of the overall layout. The last part is the output buffer of 3GHz PLL. Because we use other group's output buffer, we don't have enough information to adjust the layout. Another reason of ineffective utilization of the chip area is that the AD-SSCG, 3GHz PLL, and 3GHz output buffer is not originally integrated in the same chip.

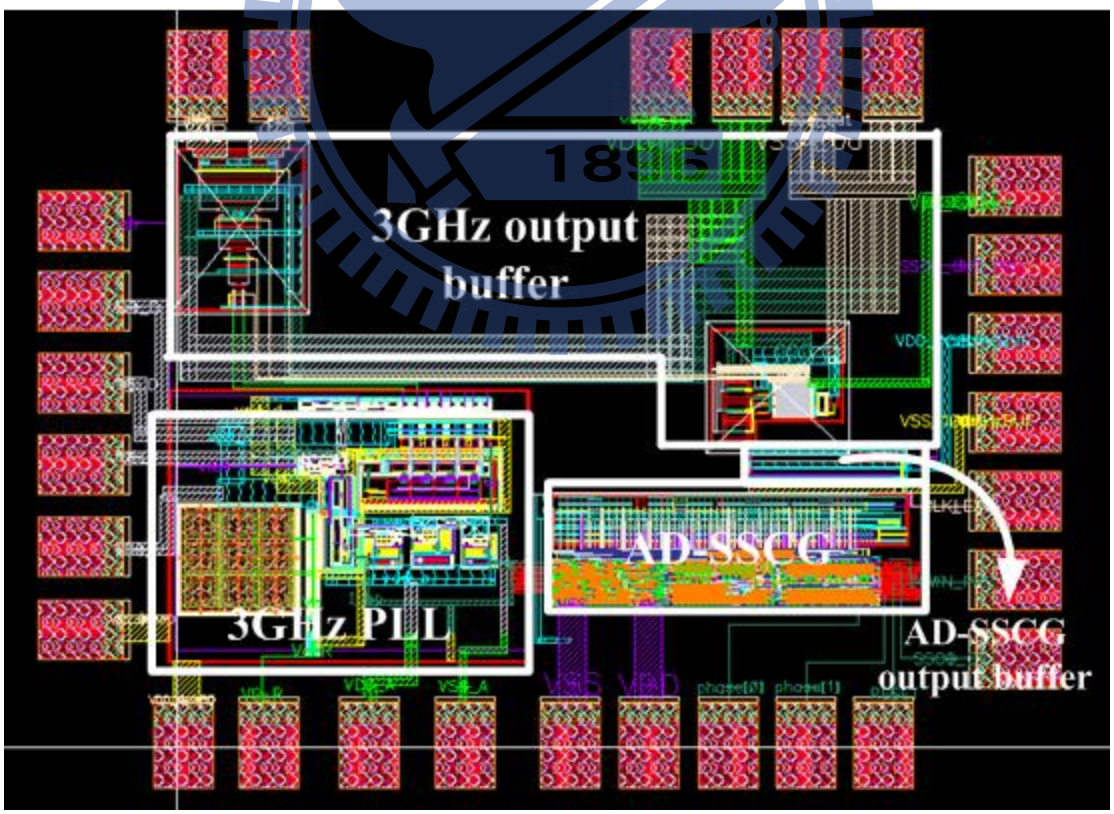


Fig. 4.34 Overall all layout of AD-SSCG and 3GPLL

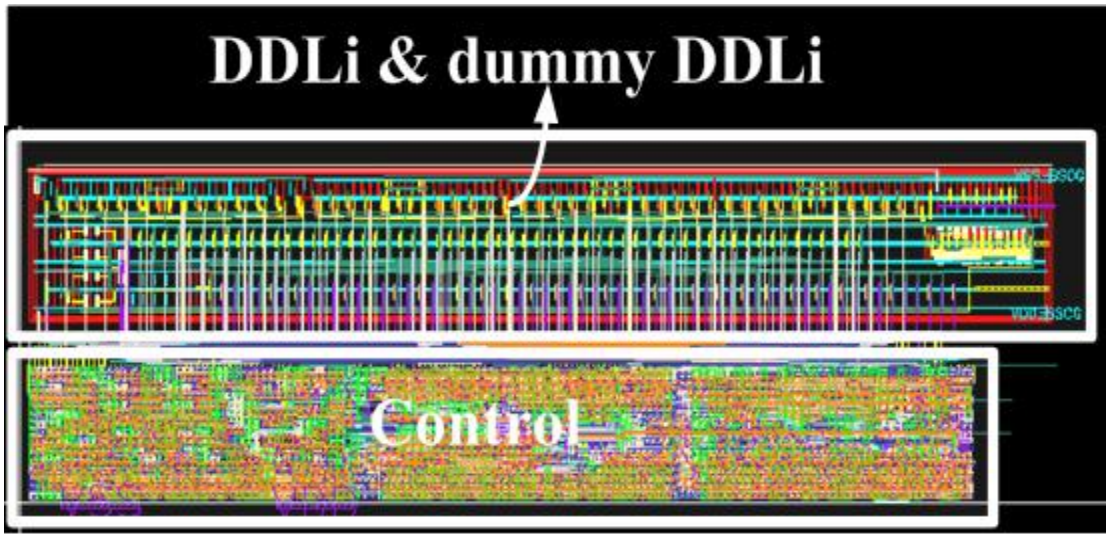


Fig. 4.35 Layout of AD-SSCG

4.7.3 Measurement Environment Setup

The testing environment setup is shown in Fig. 4.36. All DC supply sources are given from Keithley 2400 Source Meter. Agilent N4903A Serial BERT provides clock as the input of AD-SSCG. Tektronics TDS6124C Digital Storage Oscilloscope is used to measure the waveform of 3G PLL w/i and w/o spread spectrum. Agilent E4440A Spectrum Analyzer is used to observe the spectrum of AD-SSCG and 3GHz PLL.

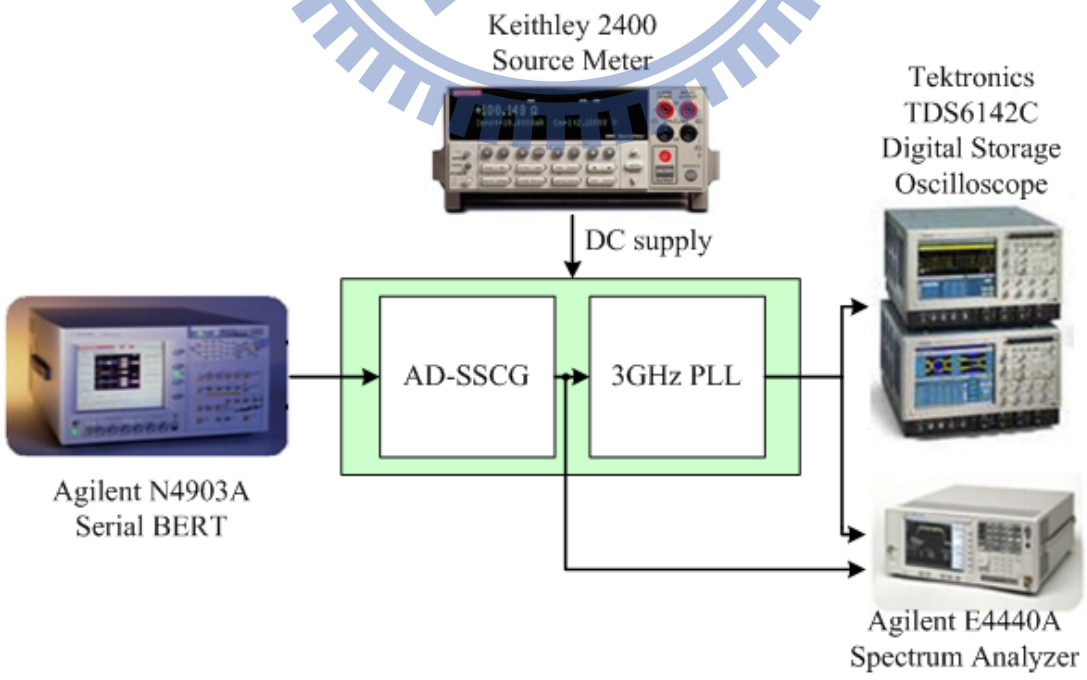


Fig. 4.36 Test environment setup

4.8 Comparison and Conclusion

In the summarized Table 4.4, we list some important and latest papers of SSCG with different modulation methods. Firstly, we present new modulation method of input reference, and compare with (2). The improvement is that we can achieve higher frequency modulated_CLK with less frequency deviation and different amount of frequency deviation in one AD-SSCG due to the proposed Domino modulation method. However, we compare with the EMI reduction/ BW, and the performance is worse than (2). The paper [16] gives a simple guideline to estimate the EMI reduction. The EMI reduction is relative with modulation frequency, operation frequency, and frequency deviation. In general, lower operation frequency, higher modulation frequency, and more amount of frequency deviation will have better EMI reduction. We can observe (2), (3), and (4), the lower operation frequency will have better EMI reduction/ Spread BW. And (1), (6), and (7) with higher operation frequency won't have better EMI reduction/ Spread BW.

Table 4.4-1 Comparison of SSCG with different modulation method

	(1) Proposed	(2) JSSC 2007 [12]	(3) CAS-I 2008 [17]	(4) JSSC 2003 [6]
Technology	90nm	0.15um	0.35um	0.35um
Modulation method	Input Reference	Input Reference	VCO	VCO
Modulation frequency	31.5KHz	100KHz	40KHz	40KHz
Modulation profile	Triangular	Triangular	Triangular	Triangular
Modulation mode	Up/Down	Up/Down	Center	Center
Operation frequency	3GHz /100MHz	148.5MHz /27MHz	400MHz	66/ 133/ 266 MHz
Frequency deviation	0.5/1.0/1.5%	3%	0.5/ 1.0/1.5/ 2%	0.5/ 1.0/ 1.5/ 2.0/ 2.5%
EMI reduction	21.5dB (3GHz,0.5%) Simulated	13dB (148.5MHz,3%) Measured	16.3dB (400MHz,1.5%) Measured	12dB (266MHz,2.5%) Measured
Spread Bandwidth	15MHz	4.455MHz	6MHz	6.65MHz
EMI reduction /Spread BW	1.48dB/MHz	2.92dB/MHz	2.72dB/MHz	1.8dB/MHz
Power	2.9mW (w/o PLL) 57.4mW (w/i PLL)	7.1mW (w/o PLL) N.A. (w/i PLL)	27.5mW	300mW

Table 4.4-2 Comparison of SSCG with different modulation method

(5) ISSCC 200 [11]	(6) CICC 2006 [10]	(7) CAS-Ⅰ 2009 [18]	(8) CAS-Ⅱ [9]
0.18um	90nm	0.18um	0.18um
MUX	MUX	Divider	Divider
30KHz	31.25KHz	33KHz	31KHz
Triangular	Triangular	Triangular	Triangular
Down	Down	Up/Down	Down
1.5GHz	3GHz	2.4GHz	1.5GHz
5000ppm	4200ppm	3700ppm	5000ppm
9.8dB (1.5GHz,0.5%)	9.7dB (3GHz,0.42%)	11.4dB (2.4GHz,0.37%)	14.77dB (1.5GHz.0.5%)
7.5MHz	12.6MHz	8.88MHz	7.5MHz
1.31dB/MHz	0.78dB/MHz	1.28dB/MHZ	1.97dB/MHz
N.A.	83.6mW	36mW	34.2mW

The contribution of paper [12] is that although the generated modulated_CLK whose number of steps in maximum frequency deviation is not enough to have better EMI reduction. But when the generated modulated_CLK is feed into a PLL, the spread spectrum function of PLL is better than original modulated_CLK. We collect some papers that also can use their modulated_CLK to be tracked by a PLL. Table 4.5 summarizes the information.

If paper [19] and [20] use the generated lower frequency modulated_CLK to be tracked by a PLL, it can have multiplied frequency and have better spread spectrum

function. Comparing to these three papers, we can generate higher frequency modulated_CLK and the modulated_CLK have more steps in maximum frequency deviation. The proposed Domino AD-SSCG can be widely used for many applications. Because (1) it can generate higher frequency modulated_CLK as shown in Table 4.5. Comparing to higher frequency structure of DDLi, the required delay of BUF for lower frequency structure of DDLi is easy to design in our design. (2) More steps in maximum frequency deviation will make the PLL have good triangular profile and EMI reduction.

Table 4.5 Comparison of SSCG with input reference modulation

	Proposed	JSSC 2007 [12]	ISCAS 2008 [20]	IEICE 2005 [19]
Technology	90nm	0.15um	65nm	0.15um
Frequency deviation	0.5/ 1.0/ 1.5%	3%	5%	0.5%
Modulation profile	Triangular	Triangular	Triangular	Triangular/ Saw
Modulation mode	Up/Down	Up/Down	Center	Down
modulated_CLK	100MHz	27MHz	33/100/ 133MHz	25MHz
Numbers of steps	40	6	10	8
EMI reduction (before PLL) (after PLL)	 21.5dB(0.5%)	N.A. 13dB(148.5MHz)	7dB(133MHz)	6.02dB (Tri) 8.02dB(Saw)
Power	2.9mW	7.1mW	1.15mW	N.A.

Chapter 5
All Digital Programmable Gaussian Clock Generator

5.1 Introduction

In high speed link, we always hope the data we received achieve high reliability. For this purpose, in the receiver end, we need ideal clock to sample the data and have data recovered by CDR (Clock Data Recovery) [21-22]. In paper [23], it describes clock and data with different Gaussian distribution will lead the system to have different BER (Bit Error Rate). In order to verify the BER of system, we often need equipment that can generate clock and data with different Gaussian deviation to verify the BER of system or circuit.

Our group previously developed all digital oversampling CDR [24], and established the analysis theory to estimate the BER of oversampling CDR [23]. In order to verify the BER of this CDR in pre-tape out procedure, we use Matlab to generate clock and data with different Gaussian deviation. However, due to the limitation of computer memory and hardware, only approximated 10^6 data can be verified in nowadays computer platform. However it can't achieve industry specification which needs at least 10^{12} testing data. So, we present a new solution. Firstly, we design a Gaussian clock generator which has programmable ability of different deviation Gaussian clock. Then, we implement Gaussian clock generator on FPGA platform. The advantage of this solution are (1) the real circuit can generate high speed operation Gaussian clock (2) there is no limitation of storage of data amounts because the platform isn't on computer (3) we can know the feature of this CDR before tape out.

Section 5.2 will describe the basic concept of digital Gaussian clock generator and

show the brief Gaussian clock generator. The Gaussian clock generator is comprised of two main parts, and we will describe the detail circuits in section 5.3 and 5.4 respectively. In section 5.3, Box-Muller algorithm will be introduced to design the Gaussian noise generator. Then, section 5.4 will show the transformation circuit which can transform a Gaussian noise to a Gaussian clock, and it is designed by all digital circuit. Section 5.5 will discuss the combination of these two circuits which are described in section 5.3 and 5.4. Also, we provide design parameters for generating different Gaussian clock will be outlined. Section 5.6 will show the experimental results on FPGA platform.

5.2 The Concept of Digital Programmable Gaussian Clock Generator

In thesis [23], it uses MATLAB to generate digital Gaussian pattern to verify the BER of CDR in verilog platform. However, the amount of data verification has some limitation. So, we present another solution, that is, we design a circuit that can generate Gaussian clock. Then, we transform the verification platform from computer to FPGA board. Therefore, it can achieve higher speed of verification time and solve the limitation of data storage at the same time.

This Gaussian clock generator is implemented by digital circuit. The reasons are (1) we want to have programmability of Gaussian jitter, (2) the circuit can be implemented on FPGA board, and we don't need to tape out, and (3) digital Gaussian clock generator can be integrated easily with digital CDR.

The concept of digital Gaussian clock generator is that we use higher frequency clock to generate lower frequency clock, but it will have Gaussian property. We call the lower frequency clock "Gaussian CLK". The function of generating Gaussian CLK is like to do division of higher frequency clock. We can control the rising edge divided clock to appear at specified position. As shown in Fig. 5.1, (a) is higher

frequency clock (fo) in system and (b) is the lower frequency clock (fd) which is divided by higher frequency clock with division ratio of 128 with 50% duty cycle (c) is the fd, but the rising edge of this clock will appear at position #63. The rising edge can be assigned at other position that we want. In (b) we assign the clock number in fo is 63. When the assigned clock number is changed in fo, the position of rising edge of fd is also changed. We use many unit period of fo to exchange that the rising edge of fd can happen at different positions. When the assigned number is a Gaussian random variable, the rising edge will appear with Gaussian property as shown in Fig. 5.1 (d). Our design goal is to generate this kind of Gaussian CLK.

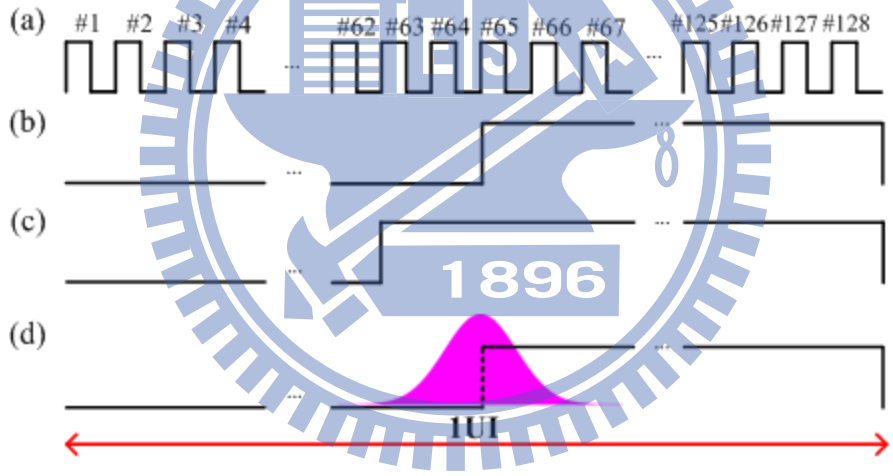


Fig. 5.1 (a) higher frequency clock (fo), (b) normal divided (fd) clock with 50% duty cycle,(c) divided clock (fd) with rising edge at #63, and(d) divided clock (fd) with assigned number is a Gaussian random variable.

In Fig. 5.1 (d), we define 1 UI is one period of Gaussian CLK (fd). The division ratio of this example is 128. If the division ratio is larger, the Gaussian CLK will be more similar like real Gaussian clock. But the generated Gaussian CLK will be slower. The division ratio is a design consideration

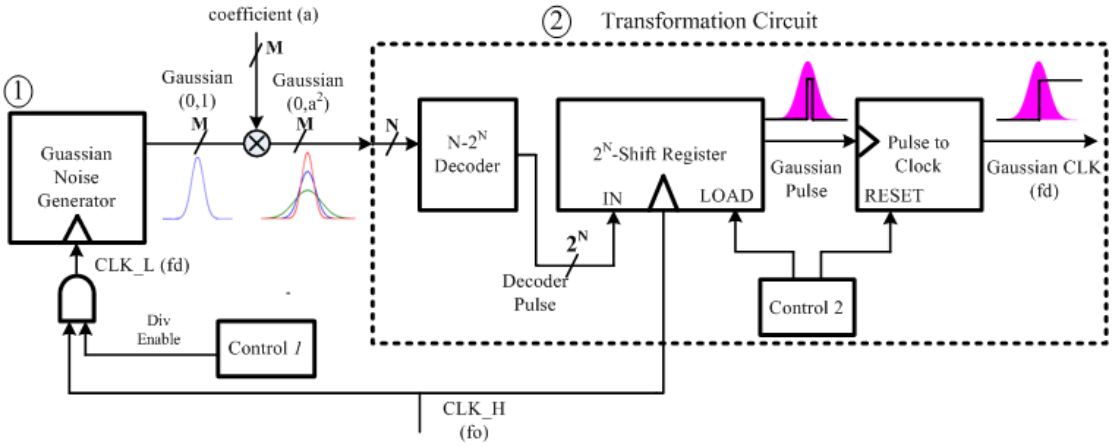


Fig. 5.2 The brief architecture of programmable Gaussian clock generator

There are two parts in Gaussian clock generator, and the brief circuit architecture is shown in Fig. 5.2. (1) "Gaussian Noise Generator" circuit (introduced in section 5.3), and (2) is the "Transformation Circuit" which can transform a value to a clock whose position of clock rising edge will appear depends on the magnitude of the value (introduced in section 5.4). We can see Fig. 5.2, firstly, "Gaussian Noise Generator" will generate normal Gaussian random variable (0,1). The first and second value respectively means the average value and the variance of Gaussian random variable. After generating normal Gaussian random variable, we will multiply a coefficient to normal Gaussian random variable to change the variance to meet the desired Gaussian distribution. Then, we introduce the transformation circuit. The "N-2^N Decoder" will receive the Gaussian value, and generate the signal which is includes the magnitude of Gaussian value. Because the shape of "N-2^N Decoder" output is like pulse (00…1…0), Here, call it "Decoder Pulse" in Fig. 5.2. The position of "1" will depend on the received value. The "Decoder Pulse" signal will be loaded into "2^N Shift-Register". The output of "2^N Shift-Register" needs 2^N CLK_H to generate a pulse signal which has Gaussian property. The final circuit "Pulse to Clock" is used to transform a pulse signal to a clock signal. Then, we can get a Gaussian CLK. This all circuit is operated at two different frequency. Beside CLK_H, there is the other frequency of clock in

circuit, that is, CLK_L. And the CLK_L is divided by CLK_H. “” Gaussian Noise Generator” is operated at CLK_L. Comparing to the concept of Gaussian clock in Fig. 5.1, the frequency of CLK_H is fo and the frequency of CLK_L and the generated Gaussian CLK is fd.

5.3 Gaussian Noise Generator

5.3.1 Gaussian Algorithm of Box-Muller

Box-Muller (BM) [25] is an algorithm that can use uniform random variable U(0,1) (duration within 0 and 1), LUT (Look Up Table), and multiplication to perform normal Gaussian random variable (zero mean and variance of 1). The advantage of BM algorithm is the less computation complexity. Thus, it is suitable for hardware implementation. The uniform random variable U(0,1) is performed by LFSR (Linear Feedback Shift Register). The LUT saves pre-computed value of $f(x_1)$ and $g(x_2)$ in equation (5.1) and (5.2). Finally, we multiply $f(x_1)$ and $g(x_2)$ together to generate function $h(x1,x2)$ that is normal Gaussian random variable in equation (5.3).

$$f(x_1)=\sqrt{-2\ln(x_1)} \tag{5.1}$$

$$g(x_2)=\sin(2\pi x_2) \tag{5.2}$$

$$h(x1,x2)=f(x_1)\times g(x_2) \tag{5.3}$$

where x_1 and x_2 are independent uniform random variable U(0,1). h is normal Gaussian random variable.

The ideal Gaussian random variable h is distributed between $-\infty$ to $+\infty$. However, the hardware implementation will restrict h to have limited distribution. If we want function h has wider distribution, LUT of $f(x_1)$ will consume a lot of area. For example, when we use 12 × 12 bit uniform LUT of $f(x_1)$, $f(x_1)$ will distribute between 0 to 4.079. We know the maximum and minimum value of $g(x_2)$ is +1 and -1, so the multiplication of $f(x_1)$ and $g(x_2)$ will distribute +4.079 to -4.079. Paper [26] adopts non-uniform LUT algorithm of $f(x_1)$ to save the storage requirement for

f (x_1), and have reasonable accuracy at the same time. Using algorithm of paper [24], the LUT of g (x_2) will become five 4 × 10 bit LUT with the same distribution. F (S_1) is the hardware implementation function of f (x_1) and it adopts the non-uniform LUT algorithm [24] as shown in equation (5.4).

$$F(S_1)=R[2^M f(\frac{S_1-\delta}{16^r})] \tag{5.4}$$

where S_1 is four digital bits, it varies between 1 and 16 (segment number). δ is a real number between 0 and 1, it can adjust the variance of Gaussian random variable. F (S_1) is coded with (3,M) bits, where the first and the second part is the number of bits in integer and fraction. The non-uniform LUT is a recursive algorithm where **r** defines the repeated times of F (S_1).

Here, we introduce the non-uniform quantization method which is obtained by a recursive partition of segment. One segment will be partitioned in 16 sub-segments with the same length. The first segment is [0,1], and divided into 16 small segments. Then, the second segment is [0,1/16], and divided again into 16 small segments and so on. The partition operation is performed r times.

G (S_2) is the hardware implementation function of g (x_2) as shown in equation (5.5). It adopts uniform LUT algorithm.

$$G(S_2)=R[2^N \sin(\frac{S_2\pi}{512})] \tag{5.5}$$

where S_2 is 10 digital bits, it varies between 1 and 1024. G (S_2) is coded with (1,N) bits, where the first part is sign bit and the second part is the number of bits fraction.

$$H(S_1,S_2)=F(S_1)\times G(S_2) \tag{5.6}$$

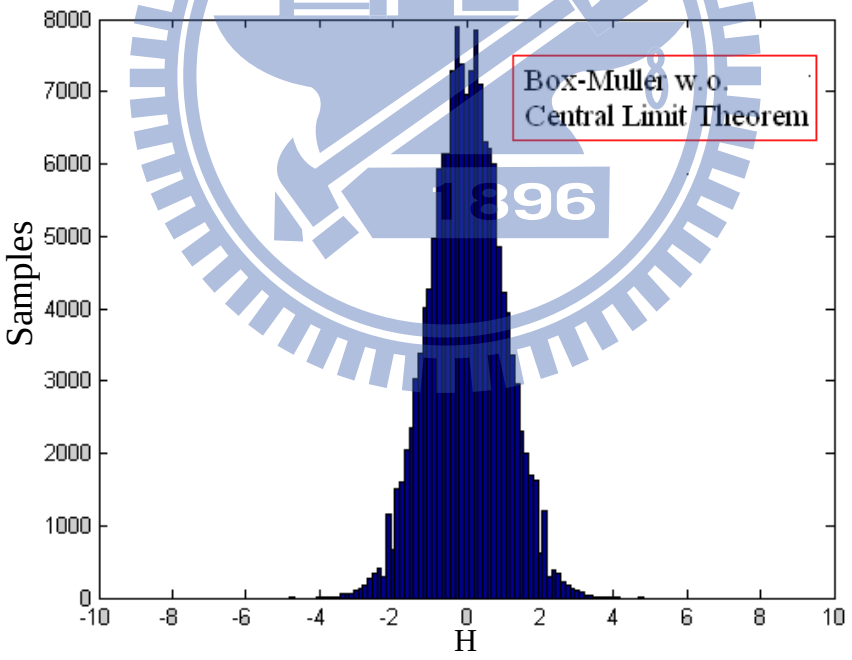
H is the function Gaussian random variable with hardware implementation as shown in equation (5.6).

Although non-uniform quantization will save a lot of storage of LUT, it will make Gaussian random variable have some distortion as shown in Fig. 5.3 (a). We can use

Central Limit Theorem [27] to amend the distortion due to using non-uniform LUT. The equation (5.7) is Central Limit Theorem, where n_p is independent random variable with zero mean and variance of 1.When p → ∞, $\bar{n}$ will become normal Gaussian random variable. We find that p=4 is sufficient to amend the less distortion, so we accumulate H (S_1,S_2) four times to complete a normal Gaussian random variable as show in equation (5.8). The amended results is shown in Fig. 5.3 (b), the numbers of samples is 150000. Besides, when p is 4, the division ratio of 2 is also easy to implement in hardware.

$$\bar{n} = \frac{n_1 + n_2 + ... + n_p}{\sqrt{p}} \tag{5.7}$$

$$\overline{H} = \frac{H_1 + H_2 + H_3 + H_4}{\sqrt{4}} \tag{5.8}$$



(a)

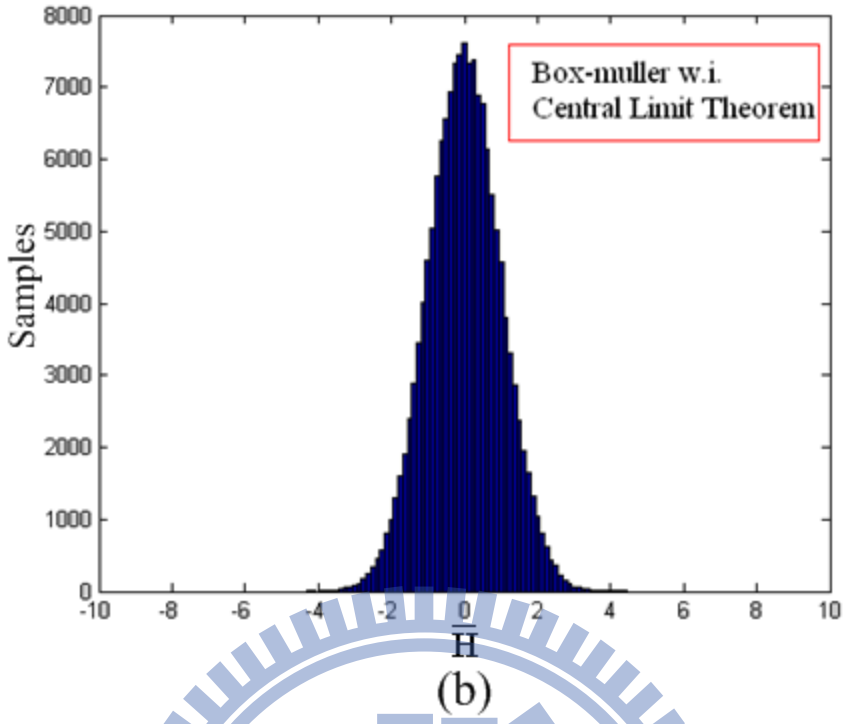


Fig. 5.3 150000 samples with non-uniform LUT Box-Muller algorithm

(a) w.o. and(b) w.i. Central Limit Theorem.

Now, we have already generated the normal Gaussian random variable. However, we need to generate different Gaussian noise with different standard deviation to verify different environment. We can use Gaussian random variable transformation [27], that is, when $\overline{H}$ multiplies coefficient, the standard deviation of Gaussian random variable will be changed as shown in the equation (5.9). The implementation of hardware is just to multiply a coefficient to normal Gaussian random variable.

$$\begin{aligned} &\text{if} && \overline{H} \sim Ga \\ &\text{let} && \widetilde{H} = a \times \overline{H} \\ & && \widetilde{H} \sim \text{Gauss}(0, a^2) \end{aligned} \tag{5.9}$$

where $\overline{H}$ is a Gaussian random variable with zero mean and variance of 1 and the multiplied values a is a real number. $\widetilde{H}$ is a Gaussian random variable with zero mean and variance of a^2.

5.3.2 The Modified Box-Muller Algorithm for CDR Application

In paper [24], it claims the distribution of generated Gaussian random variable is between -4 to +4 deviation. However, the Gaussian clock for CDR testing should

have wider value, the generated Gaussian clock should like ideal Gaussian clock which can distribute between $-\infty$ to $+\infty$. If the Gaussian clock could present wider distribution, the simulation for CDR system is more like practical testing environment. We need to design at least a Gaussian clock generator which can distribute from -7 to +7 deviation to cover 10^{12} cases. So, we extend the concept of non-uniform algorithm [26] to make LUT which can have wider distribution and consume reasonable area. The work in [26] adopts 5 banks (r=5) of non-uniform LUT. In our design, we increase the banks to make Gaussian random variable H has wider value. Finally, the numbers of banks are extended from 5 to 9 (r =9). The hardware overhead is four 4 × 10 bits LUTs, and the Gaussian random variable H can distribute between -7 to +7. The parameters of the final Gaussian random variable H and Box-Muller respectively show in Table 5.1 and 5.2.

Table 5.1 The parameters of Gaussian random variable $\overline{H}$

mean	variance
-0.0029	0.9990

Table 5.2 The parameters of Box-Muller algorithm

r	δ
9	0.53

5.3.3 Hardware Implementation of Gaussian Noise Generator

Fig. 5.4 is the overall architecture of Gaussian Noise Generator. LFSR will generate random variable U(0,1). The SIN table uses uniform quantization. The LOG table uses non-uniform quantization. The output of multiplier is 20 bits, and we quantize the results to 10 bits. Accumulator adds the output of multiplier for four times to implement Central Limit Theorem, the input are n(i), n(i+1), n(i+2), and n(i+3) respectively. ACCU output is a Gaussian (0,1) random variable. Then the multiplied coefficient will let us have Gaussian random variable $(0,a^2)$ with required standard deviation, and it also embed the division ratio of 2 of Central Limit Theorem. Finally, the final quantized Gaussian random variable is 10 bits. The use of finite word length will reduce the area of hardware.

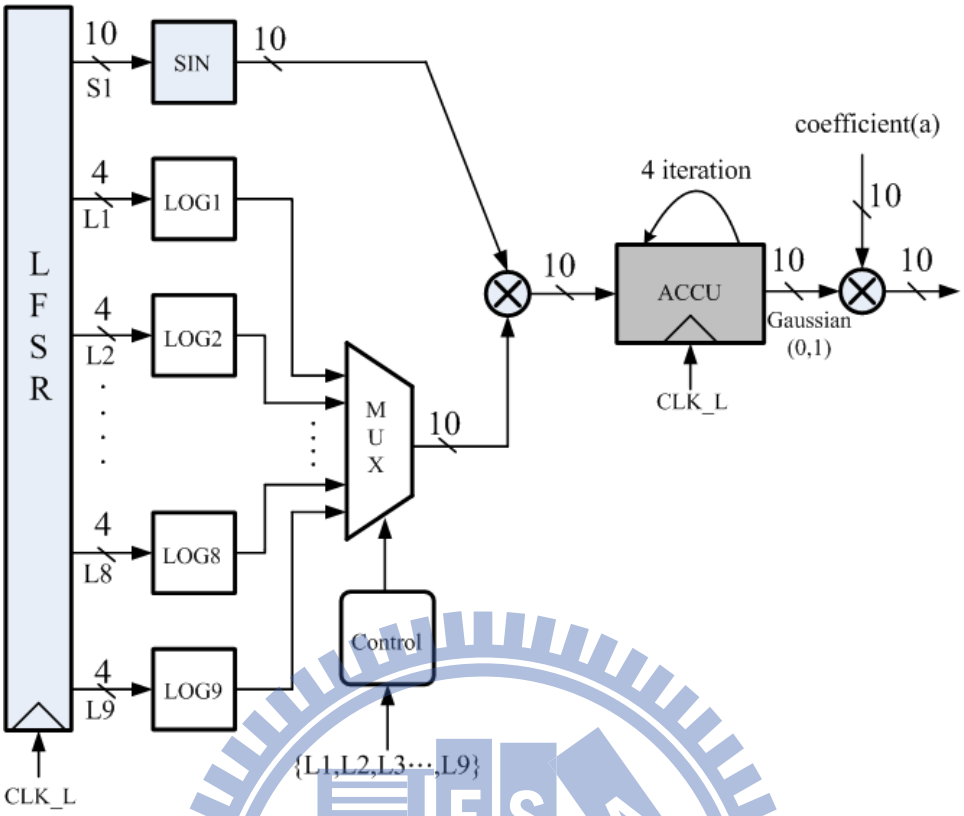


Fig. 5.4 Overall architecture of Gaussian Noise Generator

5.4 Transformation Circuit of Gaussian Noise to Gaussian CLK

We design a circuit that can transform a Gaussian value to a Gaussian CLK whose position of clock rising edge is relative to Gaussian value. The "Gaussian noise generator is in front of the transformation circuit. When the transformation circuit receives a larger value which is generated by Gaussian noise circuit, the rising edge of Gaussian CLK will be triggered later as shown in Fig. 5.5 and vice versa.

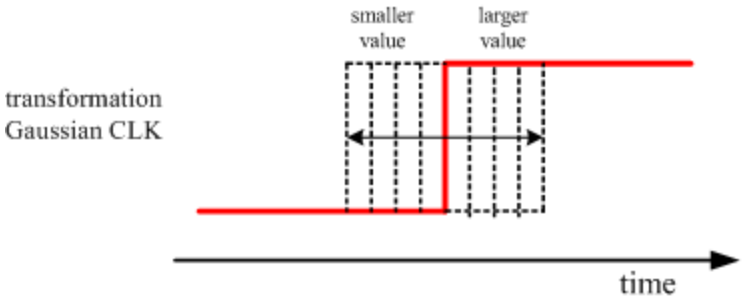
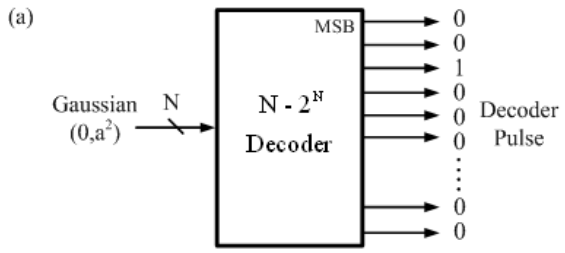


Fig. 5.5 The relative position of clock rising edge with different value

We will use digital circuit to design a circuit that can transform Gaussian value to a Gaussian CLK. Firstly, we will use the feature of decoder. When we input a value to a decoder, the output will be 001000…00 as shown in Fig. 5.6 (a). There is only "1"

in many of the output bits. The position of the "1" is in the K-th position if value of the input value number is K. Thus, we find that the position of "1" will depened on the magnitude of input value. Besides, the bit stream of decoder output bits is like a pulse signal as shown in Fig. 5.6 (b). Here, we call the bit stream as "Decoder Pulse". The input of N-2^N Decoder is generated by Gaussian noise generator. Next, we will let the "Decoder Pulse" signal becomes the function which is relative with time. Next circuit will complete this function.

Then, we use shift-register to let the "Decoder Pulse" signal become the function which is relative with time. As shown in Fig. 5.7, we connect "Decoder Pulse" of N-2^N Decoder to the IN of 2^N Shift-Register. We set the length of Shift-Register is the same as Decoder output. At 1st CLK_H cycle, we load the "Decoder Pulse" signal into 2^N Shift-Register. During 2nd to 2^{N-th} CLK_H cycles, the "Decoder Pulse" signal will be sent out one by one. After 2^N clock cycles, a complete "Decoder Pulse" signal will be send out from the output of 2^N Shift-Register. If the input value of Decoder is larger, the "1" will appear late in the output of Shift-Registers. If the input value of Decoder is smaller, the "1" will appear early in the output of Shift-Register. Fig.5.8 shows an example, assuming that every 8 CLK_H will generate a pulse signal. We can observe that when the input value of 3-8 Decoder is larger, the pulse signal will be triggered late at 6-th CLK_H as shown in Fig. 5.8 (b). And when the input value of Decoder is smaller, the pulse signal will be triggered early at 3-rd CLK_H as shown in Fig 5.8 (b). We use Shift-Register to save the information of "Decoder Pulse", and every CLK_H will trigger Shift-Register to send out a part of "Decoder Pulse". So, this pulse signal will the function of time due the feature of Shift-Register.



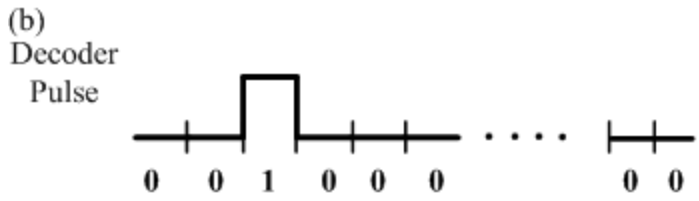


Fig. 5.6 (a) N-2^N Decoder (b) The "Decoder Pulse" of N-2^N Decoder

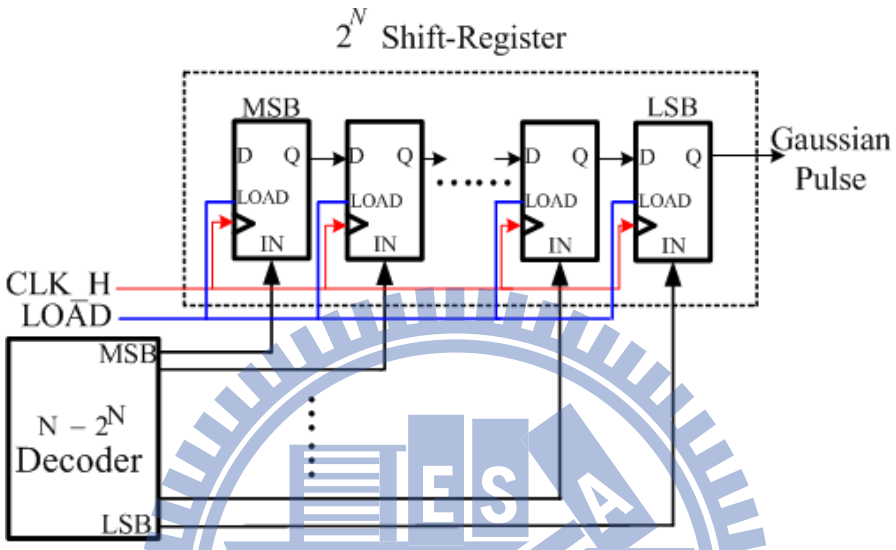


Fig. 5.7 The connection between N-2^N Decoder to 2^N Shift-Register

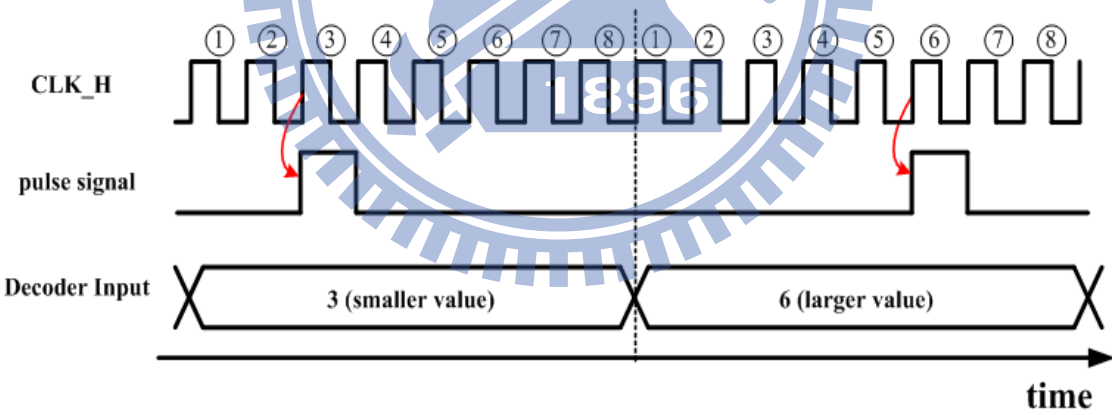


Fig. 5.8 An example of pulse signal with different input of Decoder Input

Finally, we use "Pulse to Clock transformation circuit" to transform the pulse signal to a clock signal. The concept of transforming from a pulse signal to a clock signal is shown in Fig. 5.9 (a). We can use simple DFF and an extra "Reset Enable" signal to generate this function as shown in Fig.5.8 (b). The input D of DFF is connected to VDD. The input CLK of DFF is connected to "Gaussian Pulse" which is generated by 2^N Shift-Register. The input RESET of DFF is connected to "Reset

Enable". When the DFF is triggered by the rising edge of "Gaussian Pulse", the output Q of DFF will become "1". Then, an extra "Reset Enable" will reset the DFF output of Q to "0". The triggering time has the property of Gaussian property, and the reset time is a fixed time. So, we can transform a pulse signal to a clock signal whose rising edge has the property of Gaussian distribution. The time diagram of "Pulse to Clock transform circuit is shown in Fig. 5.8 (c).

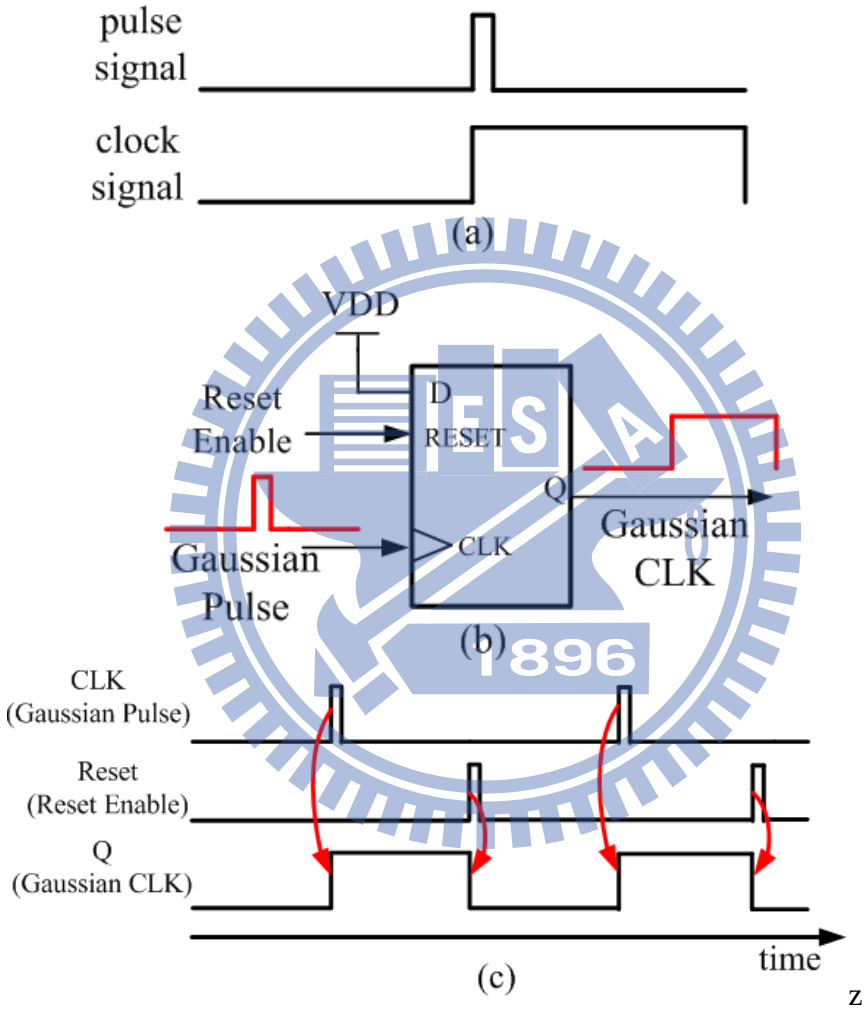


Fig. 5.9 (a) Pulse signal and clock signal (b) Pulse to Clock transformation circuit (c) The time diagram of pulse to clock transformation circuit

We combine these three digital circuits to perform the function that can transform a value to a relative position of clock rising edge. The value is generated from "Gaussian Noise Generator", so the output of "Transformation Circuit" will be a Gaussian CLK.

5.5 Circuit Implementation of Programmable Gaussian Clock Generator

We integrate "Gaussian Noise Generator" and "Transformation Circuit" to form a Gaussian clock generator. During the procedure of integrating, we have to take care of some design details. (1) In transformation circuit, Shift-Register needs 2^N CLK_H to deliver a complete pulse signal. However, Gaussian Noise Generator only needs one CLK_H to generate a noise sample. It means if we want to integrate these two circuits, there are two clock domains in Gaussian clock generator. So, we need some mechanism to solve this problem. (2) We divide higher frequency clock (fo) to generate a lower frequency Gaussian clock (fd), and we will discuss structures of different division ratio and its feature. (3) When we transform a Gaussian value to a Gaussian CLK, we need to estimate the relation between the standard deviation in Gaussian random variable and Gaussian CLK.

The division ratio is 128 in our design. When "Gaussian Noise Generator" circuit generates a noise sample, "Transformation Circuit" needs 128 CLK_H to perform a Gaussian CLK. There are two clock domains in this architecture. In Fig. 5.10, "Gaussian Noise Generator" use an AND gate and "Div Enable" signal to block CLK_H, and it will be triggered one time during 128 CLK_H. Here, we call the divided lower frequency clock as "CLK_L". The "Gaussian Noise Generator" is operated at frequency of CLK_L. And the "Transformation Circuit" is operated at frequency of CLK_H.

The division of 128 is designed for our requirement. We also can choose different division. If the division ratio is 256, we will get lower frequency Gaussian CLK than the one with division ratio of 128, but the Gaussian CLK will have better resolution.

So, the division ratio has a trade-off between speed and resolution. Fig. 5.11 shows an example with different division ratio. We can observe the higher division ratio will generate to slower Gaussian CLK, but have better resolution. Here, we can use Fig. 5.11 to explain the resolution we define more clearly. The rising of Gaussian CLK with division ratio of 16 can appear from #1 to #16. The resolution is 1/16 UI. The rising edge of Gaussian CLK with division ratio of 8 only can appear from #1 to #8.

The resolution is 1/8 UI. If the resolution is good enough, the digital Gaussian CLK will be like analog Gaussian CLK

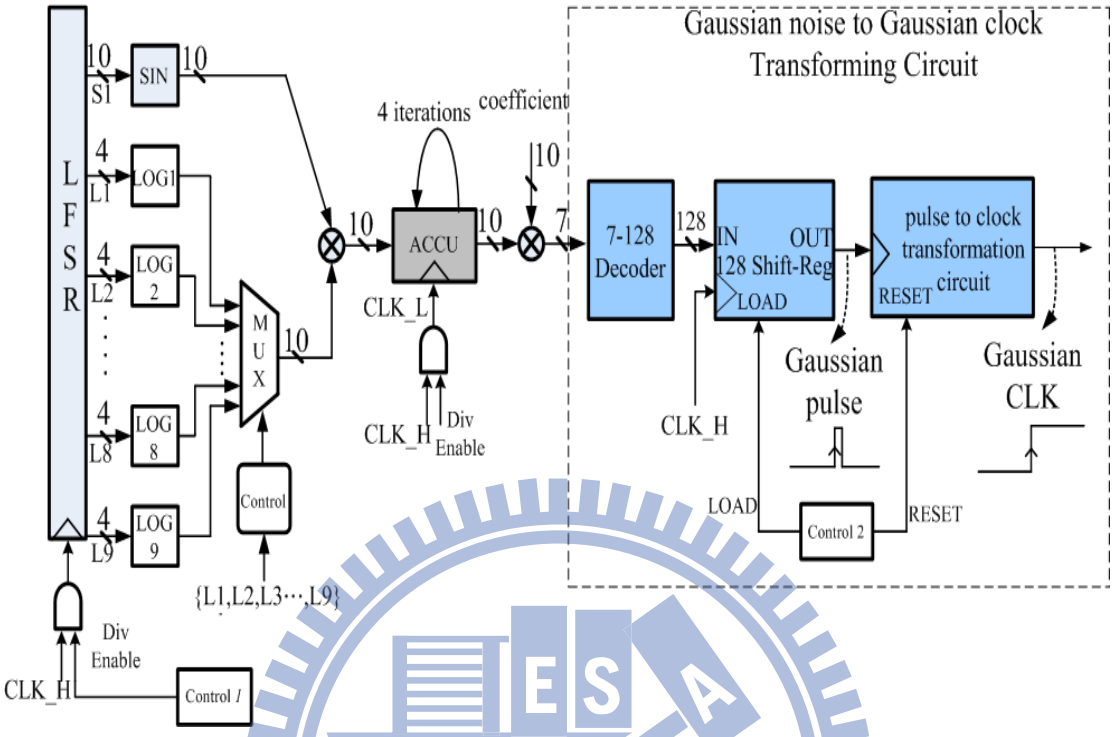


Fig. 5.10 Overall architecture of Gaussian clock generator circuit

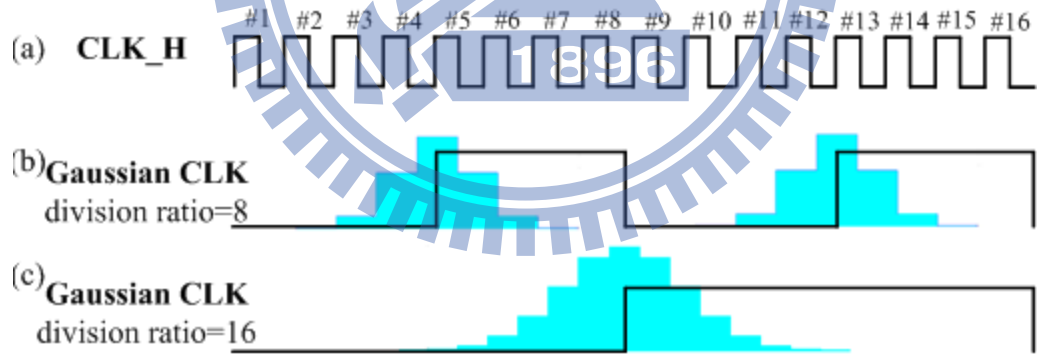


Fig. 5.11 Gaussian CLK example with different division ratio.
(a) higher frequency CLK_H, (b) Gaussian CLK with division ratio = 8,
(c) Gaussian clock with division ratio = 16.

Table 5.3 shows that when we choose different division ratio, and we need to have different structures of Decoder and Shift-Register. Also the "Div Enable" signal in Fig. 5.10 will be changed when you choose different division ratio. For example, if the division is ratio 64, the "Div Enable" will enable once during 64 CLK_H.

Table 5.3 Different hardware structures with different division ratio

division ratio	Decoder	Shift-Register
64	6-64	64
128	7-128	128
256	8-256	256

The value generated by "Gaussian Noise Generator" is between -7.984375 ~ +7.984375 (coded with (1,3,6) bits), where the first part is sign bit and the second and the third is the number of bits in integer and fraction, and its standard deviation is 1. Now, we transform this value into one clock (1 UI), the standard deviation of clock will become 0.0625UI (1/16=0.0625). It means when we generate a Gaussian random variable with zero mean and variance of 1, the relative Gaussian CLK has zero mean and variance with 0.0625UI. When we need Gaussian CLK different standard deviation to test, we can multiply different coefficient to meet our application. Table 5.4 is the required multiplied coefficient which is relative to Gaussian CLK with different standard deviation.

Table 5.4 Different standard deviation Gaussian CLK with relative coefficient

coefficient	0.16	0.32	0.64
standard deviation of Gaussian CLK	0.01UI	0.02UI	0.04UI

Here, we show some experimental results of Gaussian Clock Generator. Fig. 5.12 is the original Gaussian Clock Generator with embedded verification circuit which is comprised of 128 counters. The Gaussian Clock Generator synthesized on FPAG development board doesn't have this part of verification circuit. When Gaussian Noise Generator generate a Gaussian value, N-2^N Decoder will generate a "Decoder Pulse". The "1" in "Decoder Pulse" means one occurrence of this Gaussian value. We connect Counter to every bit of "Decoder Pulse". When one of the "Decoder Pulse" bits becomes "1", the counter will accumulate the number. We set Gaussian Clock Generator to run 10000 samples of Gaussian value and collect the data to draw the histogram. Fig. 5.13 shows Gaussian CLK with standard deviation of 0.04UI, 0.02UI

and 0.01UI.

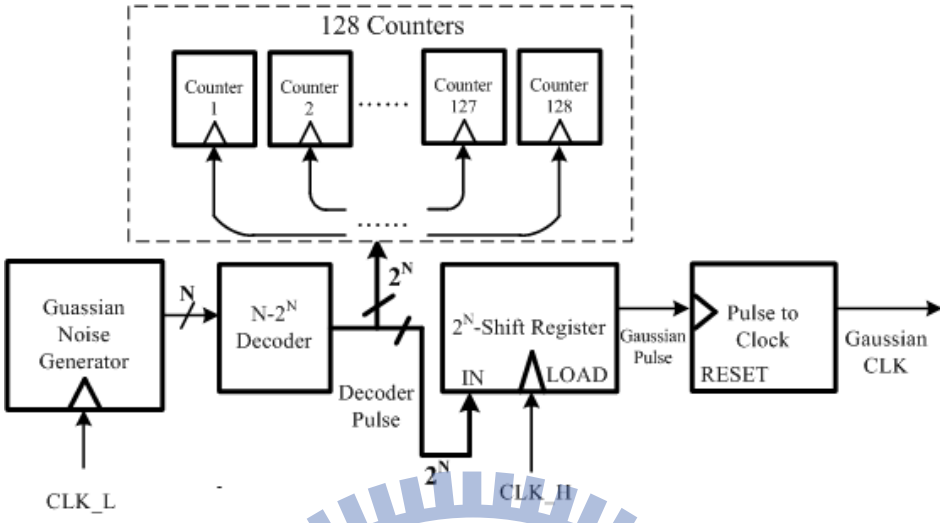
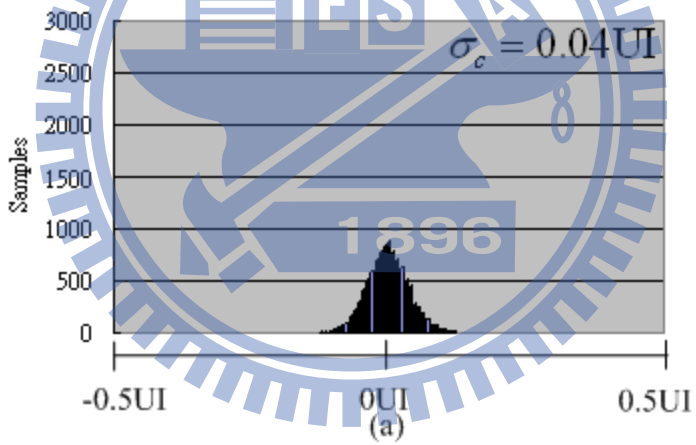
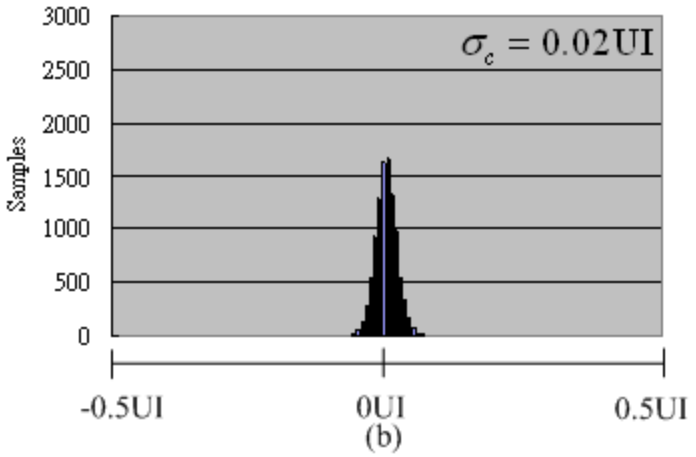


Fig. 5.12 Gaussian Clock Generator with embedded verification circuit





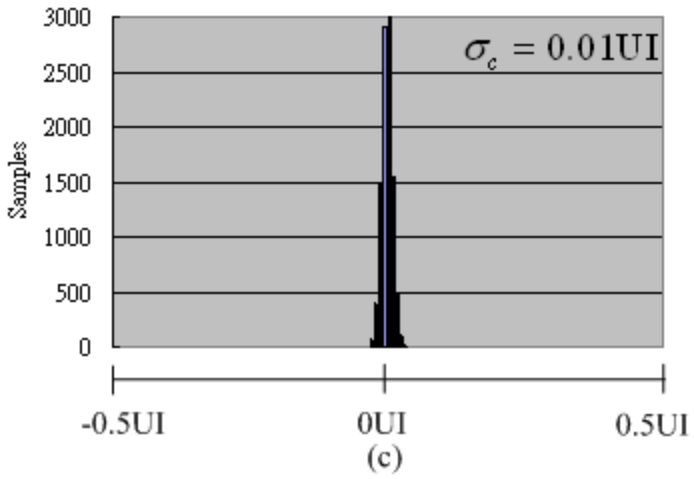


Fig. 5.13 Gaussian CLK (a) σ c = 0.04UI (b) σ c = 0.02UI (c) σ c = 0.01UI

5.6 Experimental Measurement

Fig. 5.14 shows all measured instruments. We use FPGA development board to implement programmable digital Gaussian clock generator. The series of FPGA board is EP1S80B956C6. The signal of Gaussian CLK is derived from FPGA development and Agilent infiniium 54823D MSO is used to measure Gaussian CLK. When we implement programmable digital Gaussian clock generator on FPGA, it only occupy a small percentage of FPGA logic element (2%). The crystal component in FPGA board is 8.192MHz, we use built-in PLL to synthesize 98.3MHz clock (multiplication=12) to be the clock of Gaussian clock generator. The frequency of Gaussian CLK is 0.768MHz with division ratio of 128. Table 5.4 shows the experimental setting. The clock rising edge of generated Gaussian CLK has the property of Gaussian distribution and can be different standard deviation. The programming of standard deviation can be digitally designed through a different multiplied coefficient. The speed and resolution of Gaussian CLK can be a trade-off. The higher division ratio will have lower frequency Gaussian CLK and higher resolution, and lower division ratio will have reverse property.

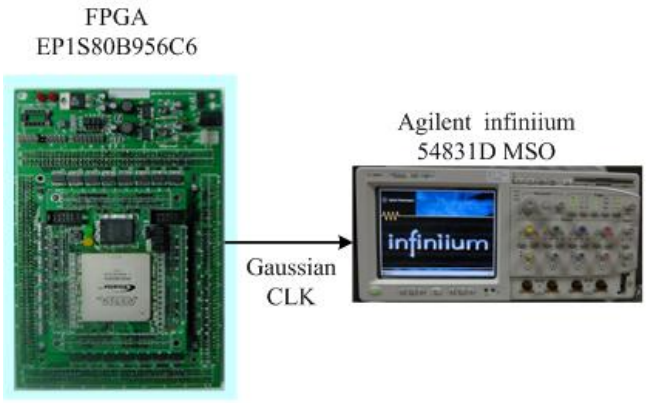


Fig. 5.14 Test environment setup

Table 5.5 Experimental results on FPGA developed board

FPGA series	Logic element	Clock Rate (MHz)	Gaussian Clock Rate (MHz)
EP1S80B956C6	1375/79040 (2%)	98.3	0.768

Aglient infiniium 54831D MSO is used to measure Gaussian CLK. The multiplied coefficient is 0.32, so it will generate the Gaussian CLK with 0.02UI. The frequency of Gaussian clock is 0.768MHz, the ideal amount of 0.02 UI jitter is around 28.1250ns. The measured Gaussian clock of 0.02UI jitter of Gaussian clock is 28.9972ns. The relative standard deviation is 0.0206UI. Fig. 5.15 shows the measurement results.

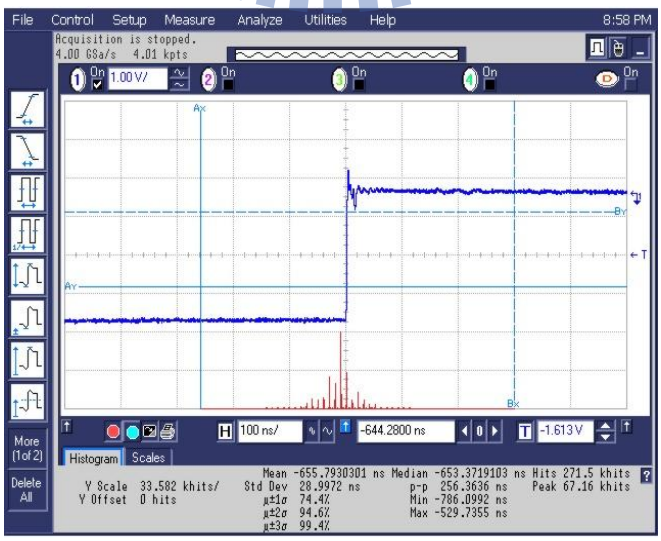


Fig. 5.15 The measurement of digital Gaussian CLK with 0.02 UI.

Chapter 6
Conclusion

We present a new SSCG method with input modulation. The proposed Domino AD-SSCG is suitable to generate higher frequency modualted_CLK with less frequency deviation. The frequency of our modulated_CLK is 100MHz which is 4 times higher than papers [12] and [17-18], and the minimum frequency deviation is 5000ppm. Besides, the Domino AD-SSCG can be operated at (1) 10000ppm and 15000ppm (2) up and down modes. The proposed AD-SSCG is very suitable for an IP. The proposed Domino AD-SSCG is simulated with a 1.2GHz and a 3GHz PLL on the MATLAB platform. The EMI reduction of 1.2GHz and 3GHz PLL with down spread 5000ppm are respectively 18.0dB and 21.5dB. And the EMI reduction of 3GHz by hspice post-sim is 22dB. In DDLi, we design a novel Coarse-Fine DDLi to improve power consumption and area utilization by 3.30 and 3.83 times. Finlay, the proposed Domino AD-SSCC and a 3GHz PLL are implemented on the same chip with UMC-90nm-CMOS 1P9M process. The EMI reduction with down spread 3GHz PLL is 22.2dB with 5000ppm frequency deviation and the area and power of this chip is 950um × 705um and 57.4mW. And the area and power of proposed Domino AD-SSCG is 335um × 105um and 2.9mW.

The novel methodology of digital programmable Gaussian clock generator. The proposed digital Gaussian clock generator can be synthesized on FPAG board, and the generated Gaussian clock can be used to verify the BER of digital CDR without tape

out. The test data will be easy up to 10^{12}. The generated Gaussian clock is designed for CDR, it can distribute from +7 to -7 standard deviation. The jitter of Gaussian clock will be change through changing the multiplied coefficient in Gaussian noise generator to simulate different test environment, like jitter with 0.01UI, 0.02UI, 0.03UI, and 0.04UI…etc. Finally, the 0.768MHz Gaussian CLK with 0.02UI standard deviation is measured, and digital programmable Gaussian clock generator only occupied a small percentage of FPGA logic element (2%).

Bibliography

[1] Ecliptek Corporation, "Programmable Spread Spectrum Quartz Crystal Oscillators Reduce EMI for High Speed Digital Systems," http://www.ecliptek.com/tech/ss-emidigi.html, Retrieved Aug. 2007,

[2] Serial ATA Workgroup, "SATA: high speed serialized AT attachment," Rev. 1, Aug. 2001.

[3] "Jitter in PLL-Based Systems: Causes, Effects, and Solutions," http://www.cypress.-com, July 1997.

[4] C.H. Chuang "A Programmable Spread Spectrum Clock Generator for Serial ATA 6Gbps," M.S. dissertation, Dep. of Electronics Engineering & Int. of Electronics, National Chiao Tung University, Taiwan, July 2006.

[5] H.S. Li, Y.C. Cheng, and D. Puar, "Dual-Loop Spread Spectrum Clock Generator," ISSCC Dig. Tech. Papers, pp. 184-185, Feb 1999.

[6] H.H. Chang, I.H. Hua, and S.I. Liu, "A Spread-Spectrum Clock Generator With Triangular Modulation," IEEE Journal of Solid-State Circuits, Vol. 38, pp. 673-676, April 2003.

[7] H.-Y. Huang, S.F. Ho, and L.-W. Huang, "A 64MHz~1920MHz Programmable Spread-Spectrum Clock Generator," IEEE ISCAS, 3363- 3366 Vol. 4, pp. 3363 -3366, May 2005.

[8] W.T. Chen, J.C. Hsu, H.W. Lune, and C.C. Su, "A Spread Spectrum Clock Generator for SATA-II," IEEE ISCAS, Vol. 3, pp. 2643-2646, May 2005.

[9] D.S. Shen and S.I. Liu, "A Low-Jitter Spread Spectrum Clock Generator Using FDMP," IEEE Trans. Circuits and Systems- II: Express Briefs, vol. 54, pp. 979-983, Nov. 2007.

[10] J. Shin, I. Seo, J.Y. Kim, S.H. Yang, C. Kim, J. Pak, H. Kim, M. Kwak, and G.-B.

Hong, “A Low-Jitter Added SSCG with Seamless Phase Selection and Fast AFC for 3rd Generation Serial-ATA,” IEEE Custom Integrated Circuits Conference (CICC), pp. 409-412, Sep. 2006.

[11] H.R. Lee, O. Kim, G. Ahn, and D.K. Jeong, “A Low-Jitter 5000ppm Spread Spectrum Clock Generator for Multi-channel SATA Transceiver in 0.18μm CMOS,” ISSCC Dig. Tech. Papers, pp. 160-161, Feb. 2005.

[12] S. Damphousse, K. Ouici, A. Rizki, and M. Mallinson, “All Digital Spread Spectrum Clock Generator for EMI Reduction”, IEEE Journal of Solid-State Circuits, Vol. 42, pp. 145-150, Jan. 2007.

[13] Y.Y. Huang “A Spread Spectrum Clock Generator for Serial ATA 6Gb/s,” M.S. dissertation, Dep. Electronics Engineering & Int. of Electronics, National Chiao Tung University, Taiwan, Dec. 2007.

[14] S.F. Al-Sarawi, “Low Power Schmitt Trigger Circuit,” Electronics Letters, Vol. 38, pp. 1009-1010, Aug. 2002.

[15] T.H. Lin “Design and Implementation of 6-Gb.s Half-Rate Clock and Data Recovery Circuit,” M.S. dissertation, Dep. of Electrical Engineering in the Graduated Division of the Int. of Electronics, National Central University, Taiwan, Nov. 2008.

[16] J. Kim, P. Jun, J.G. Byun, and J. Kim, “Design Guidelines of Spread Spectrum Clock for Suppression of Radiation and Interference from High-speed Interconnection Line,” IEEE Workshop on Signal Propagation on Interconnects, pp. 189-192, May 2002.

[17] Y.B. Hsieh and Y.H. Kao, “A Fully Integrated Spread-Spectrum Clock Generator by Using Direct VCO Modulation,” IEEE Trans. Circuits and Systems-I: Regular Papers, Vol. 55, pp. 1845-1852, Aug. 2008.

[18] C.Y. Yang, C.H. Chang, and W.G. Wong., "A Δ-Σ PLL-Based Spread Spectrum Clock Generator With a Ditherless Fractional Topology," IEEE Tran. Circuits and Systems-I: Regular Papers, Vol. 56, pp. 51-59, Jan 2009.

[19] T. Yoshikawa, T. Ebuchi, Y. Arima, and T. IWATA, "A Spread Spectrum Clock Generator Using Digital Tracking Scheme," IEICE Trans. Electron., Vol. 88, pp. 1288-1289, Jun. 2005.

[20] R. Saraswat, U. Zillmann, S. Supriyanto, G. Droege, and U. Bretthauer, "Programmable Spread Spectrum Clock Generation Based on Successive Phase Selection Technique," ISSCC Dig. Tech. Papers, pp. 2845-2848, Feb. 2008.

[21] K. Y. Chih-Kong, F. R. Ramin, and M.A. Horowitz, "A 0.5-μm CMOS 4 Gbit/s serial link transceiver with data recovery using oversampling," IEEE Journal of Solid State Circuits Vol. 33, pp 713-722, May 1998.

[22] R. J. Yang, S. P. Chen, and S. I. Liu, "A 3.125Gbps Clock and Data Recovery Circuit for the Projectsbase-LX4 Ethernet," IEEE Journal of Solid-State Circuits, Vol. 39, pp. 1356-1560, Aug. 2004.

[23] C. H. Lin, Y. I. Wang, and S. J. Jou, "Bit-Error-Rate Analysis for Clock and Data Recovery Based on Blind Oversampling Technique," International Journal of Electrical Engineering, Vol.13, pp. 209-228, July 2006.

[24] C.H. Chuang "Module Generator of Data Recovery Circuits Using Oversampling Technique," M.S. dissertation, Dep. Electrical Engineering in the Graduated Division of the Int. of Electronics, National Central University, Taiwan, June 2002.

[25] G.E.P Box and M.E. Muller, "A note on the Generation of Random Normal Deviates," Annals Math and Statistics, Vol. 29, pp. 610-611, 1958.

[26] E. Boutillon, J. L. Danger, and A. Ghazel, "Design of High Speed AWGN Communication Channel Emulator," Analog Integrated Circuits and Signal Processing, Vol. 34, pp. 133-142, Feb. 2003.

[27] H. Stark and J. W. Woods, “Probability and Random Process with Applications to Signal Processing,” Third Edition, Prentice Hall, 2001.