

# 國立交通大學

電子工程學系 電子研究所碩士班

## 碩士論文

全數位快速鎖定自我校正多相位延遲鎖定迴路設計



An All-Digital Fast-Lock Self-Calibrated Multiphase DLL

研究生：莊立溥

指導教授：黃 威 教授

中華民國九十七年七月

# 全數位快速鎖定自我校正多相位延遲鎖定迴路設計

An All-Digital Fast-Lock Self-Calibrated Multiphase DLL

研究生：莊立溥

Student : Li-Pu Chuang

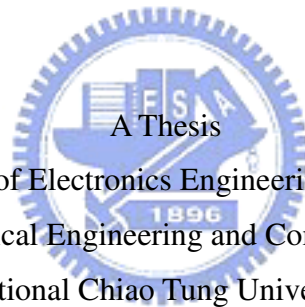
指導教授：黃 威 教授

Advisor : Prof. Wei Hwang

國立交通大學

電子工程學系電子研究所

碩士論文



A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics  
College of Electrical Engineering and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electronics Engineering

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

# 全數位快速鎖定自我校正多相位延遲鎖定迴路設計

學生：莊立溥

指導教授：黃 威 教授

國立交通大學電子工程學系電子研究所

## 摘 要

本論文提出一個全數位式快速鎖定具自我校正功能的多相位延遲鎖定迴路設計。根據所提出的快速自我校正演算法，減少因為製程不相配或是輸出負載不同造成輸出訊號的相位誤差。此外，為了達到快速鎖定，以及增加操作頻率範圍並且同時避免多諧鎖定，提出了一個非平衡式二進位搜尋演算法，其特點在於提供不同的初始延遲時間已達到上述的功能。一個非平衡式二進位搜尋控制器實現在 UMC 90nm CMOS 技術，模擬結果顯示，當延遲鎖定迴路操作頻率在 100MHz 到 500MHz (五倍)時，可以在 22 個參考時脈週期內鎖定(最差情況)。

一個 300MHz 到 1.08GHz 全數位精確多相位輸出延遲鎖定迴路實現在 UMC 90nm CMOS 技術。藉由一新型數位控制線性近似延遲元件達到線性增加延遲時間以及抗環境變異的能力。一個數位式校正單元根據所提出的快速自我校正演算法被設計與實現並且能使多相位輸出訊號的相位誤差自我校正。在校正程序結束之後，校正單元會自動關閉以減少功率消耗。在操作頻率為 500MHz 時，最大相位誤差可從 20.9ps 減少至 4.5ps。其最大消耗的總功率為 2.16 毫瓦當操作在 1GHz 時。本論文提出的延遲鎖定迴路可穩定地使用在各種嵌入式記憶體應用。

# An All-Digital Fast-Lock Self-Calibrated Multiphase DLL

Student : Li-Pu Chuang

Advisors : Prof. Wei Hwang

Department of Electronics Engineering & Institute of Electronics

National Chiao-Tung University

## **ABSTRACT**

An all-digital fast-lock self-calibrated DLL is proposed in this thesis. Base on the proposed rapid self-calibration (RSC) algorithm, the timing error caused by process mismatch and various output loading can be effectively self-calibrated. Besides, an unbalance binary search algorithm is proposed to extend the locking range and avoid harmonic lock at the same time. An unbalance binary search algorithm based (UBS) controlled is implemented in UMC 90nm CMOS technology. The simulation results show that, the operating frequency is 100MHz to 500MHz (up to 5X) and the lock-in time is down to 22 reference clock cycles in the worst case.

A 300MHz-1.08GHz all-digital multiphase delay-locked loop with precise multi-phase output has been designed with UMC 90nm CMOS technology. The linear approximate delay element property of linearity and insensitive to PVT variation is good for digitally controlled delay line. In addition, a digital calibration unit is designed based on RSC algorithm, which makes the phase error among the multiple outputs can be self-calibrated. The entire calibration unit could be turned off after calibration procedure is complete to reduce power consumption. The simulation results show the DLL exhibits a lock range from 300MHz to 1.08GHz. The maximum phase is reduced from 20.9ps to 4.5ps when the DLL is operating at 500MHz. The total power dissipation of the all-digital self-calibrated multiphase delay-locked loop is 2.16mW at 1GHz with 1V power supply. The presented DLL can be robustly used in embedded memory applications.

# Content

<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
1.1 BACKGROUND .....	1
1.2 MOTIVATION .....	1
1.3 ORGANIZATION .....	2
<b>CHAPTER 2 AN OVERVIEW OF DELAY-LOCKED LOOP .....</b>	<b>4</b>
2.1 BASIC CONCEPTS OF DELAY-LOCKED LOOP .....	4
2.2 CLASSIFICATIONS OF DELAY-LOCKED LOOP .....	5
2.3 DESIGN OF ANALOG DELAY-LOCKED LOOP .....	7
2.3.1 Voltage-controlled Delay Line.....	7
2.3.2 Phase Detector.....	9
2.3.3 Charge Pump and Loop Filter .....	10
2.3.4 Stability Analysis of Delay-Locked Loop .....	11
2.4 DESIGN OF CONVENTIONAL DIGITAL DLL.....	13
2.4.1 Register-controlled DLL [15] .....	13
2.4.2 Counter-controlled DLL [48].....	14
2.4.3 Successive Approximation Register-controlled DLL [4].....	15
2.4.4 Time measurement controlled DLL .....	17
2.4.5 Digitally Controlled Delay Line.....	18
2.4.5.1 Shunt Capacitor based DCDE .....	19
2.4.5.2 Standard Cell based DCDE .....	20
2.4.5.3 Current-starved based DCDE .....	21
2.5 COMPARISON OF DIGITAL DLL AND ANALOG DLL.....	23
<b>CHAPTER 3 MULTIPHASE DELAY-LOCKED LOOP WITH SELF-CALIBRATION.....</b>	<b>25</b>
3.1 INTRODUCTION OF MULTIPHASE DLL .....	25
3.2 APPLICATION OF MULTIPHASE DLL.....	26
3.3 SELF-CALIBRATION TECHNIQUES .....	30
3.3.1 A Self-Calibration Delay-Locked Delay Line.....	32
3.3.2 Sequential Phase Adjustment Calibration Technique.....	33
3.3.3 Parallel Phase Adjustment Calibration Technique .....	34
3.3.4 A PLL based Self-calibrated algorithm.....	36
3.4 RAPID SELF-CALIBRATION ALGORITHM .....	37

<b>CHAPTER 4 A WIDE-RANGE AND FAST-LOCK ALL-DIGITAL DELAY-LOCKED LOOP .....</b>	<b>43</b>
4.1 INTRODUCTION OF WIDE-RANGE DLL .....	43
4.2 PREVIOUS RESEARCH OF WIDE-RANGE DLL.....	44
4.3 UNBALANCE BINARY SEARCH ALGORITHM .....	46
4.4 CIRCUIT DESCRIPTION .....	50
4.4.1 Step Controller.....	50
4.4.2 Binary Controller.....	51
4.4.3 Digitally Controlled Delay Line.....	53
4.5 SIMULATION RESULTS.....	55
<b>CHAPTER 5 IMPLEMENTATION OF ALL-DIGITAL FAST-LOCK SELF-CALIBRATED MULTIPHASE DLL .....</b>	<b>57</b>
5.1 INTRODUCTION .....	57
5.1 SYSTEM ARCHITECTURE .....	59
5.2 CIRCUIT DESCRIPTION .....	60
5.2.1 Phase Detector.....	60
5.2.2 Linearly Approximant Delay Element.....	62
5.2.3 Digitally Controlled Delay Line.....	64
5.2.4 Lock-in unit.....	65
5.2.5 Calibration Unit.....	66
5.2.5.1 Digital Relative Phase Detector.....	67
5.2.5.2 Interpolator.....	69
5.2.5.3 Lock Detect Unit.....	70
5.3 DESIGN IMPLEMENTATION .....	71
5.4 SIMULATION RESULT .....	73
<b>CHAPTER 6 CONCLUSION AND FUTURE WORK.....</b>	<b>77</b>
6.1 CONCLUSION .....	77
6.2 FUTURE WORK .....	77
<b>BIBLIOGRAPHY .....</b>	<b>79</b>

# List of Tables

TABLE 1 COMPARISON OF DIFFERENT TYPE DCDE .....	23
TABLE 2 COMPARISON OF ANALOG DLL AND DIGITAL DLL .....	24
TABLE 3 SUMMARY OF THE ADSCM-DLL.....	75
TABLE 4 COMPARISON AMONG PREVIOUS WORKS.....	76



# List of Figures

FIGURE 1 THE ARCHITECTURE OF CONVENTIONAL DLL .....	4
FIGURE 2 BLOCK DIAGRAM OF ANALOG DLL.....	7
FIGURE 3 THE RCCDL (A) DELAY ELEMENT (B) DELAY LINE .....	8
FIGURE 4 THE CSCDL (A) DELAY ELEMENT (B) DELAY LINE .....	8
FIGURE 5 THREE-STATE PHASE DETECTOR.....	9
FIGURE 6 PD RESPONSES WITH (A) REFERENCE SIGNAL LAG FEEDBACK SIGNAL (B) REFERENCE SIGNAL LEAD FEEDBACK .....	9
FIGURE 7 PD STATE DIAGRAM .....	10
FIGURE 8 SIMPLE MODEL OF CHARGE PUMP AND LOOP FILTER .....	11
FIGURE 9 LOOP FILTER .....	11
FIGURE 10 SMALL SIGNAL AC MODEL OF THE CONVENTIONAL ANALOG DLL.....	11
FIGURE 11 BLOCK DIAGRAM OF DIGITAL DLL.....	13
FIGURE 12 REGISTER CONTROLLED DLL .....	14
FIGURE 13 COUNTER-CONTROLLED DLL .....	15
FIGURE 14 FLOWCHART OF 3-BIT BINARY SEARCH ALGORITHM.....	16
FIGURE 15 SARDLL.....	17
FIGURE 16 TM DLL.....	18
FIGURE 17 DCDL REALIZED BY A PATH-SELECTION METHOD .....	18
FIGURE 18 LDU AND LDL .....	19
FIGURE 19 SHUNT-CAPACITOR BASED DCDE.....	20
FIGURE 20 PARALLEL TRI-STATE INVERTER BASED DCDE.....	20
FIGURE 21 AOI-OAI PARALLEL BASED DCDE .....	21
FIGURE 22 CURRENT STARVED BASED DCDE .....	22
FIGURE 23 THE BLOCK DIAGRAM OF CONVENTIONAL DLL-BASED FREQUENCY SYNTHESIZER.....	26
FIGURE 24 THE OPERATION OF DVFS SCHEME.....	27
FIGURE 25 7:1 DATA CHANNEL COMPRESSION TRANSCEIVER. (A) TRANSMITTER CIRCUIT. (B) RECEIVER CIRCUIT. ....	29
FIGURE 26 READ OPERATION TIMING BUDGET .....	29
FIGURE 27 THE BLOCK DIAGRAM OF MULTIPHASE DLL FOR DDR SDRAM APPLICATION .....	30
FIGURE 28 DELAY TIME MISMATCH DUE TO THE DELAY CELL WITH THE THRESHOLD VOLTAGE MISMATCH OF 15 mV .....	31
FIGURE 29 DELAY TIME MISMATCH DUE TO THE DELAY CELL WITH THE CHANNEL LENGTH MISMATCH OF 10% .....	31
FIGURE 30 SELF-CALIBRATION DELAY LOCKED DELAY LINE SCHEME .....	32



FIGURE 31 SEQUENTIAL PHASE ADJUSTMENT CALIBRATION ALGORITHM .....	34
FIGURE 32 (A) RELATIVE PHASE DETECTOR (B) RELATIVE COMPARISON METHOD .....	34
FIGURE 33 PARALLEL PHASE ADJUSTMENT CALIBRATION ALGORITHM .....	35
FIGURE 34 (A) DELAY SENSING CIRCUIT. (B) CALIBRATION LOOP CHARGE PUMP .....	35
FIGURE 35 THE IDEA OF PROPOSED RSC ALGORITHM .....	37
FIGURE 36 STEP SUMMARY OF RSC ALGORITHM.....	39
FIGURE 37 THE CALIBRATION PRODUCE COMPARISON WITH (A) IQ-STYLE (B) BUFFER-STYLE (C) RSC-BASED .....	41
FIGURE 38 CALIBRATION CYCLES COMPARISON .....	42
FIGURE 39 HARMONIC LOCKING PROBLEMS.....	44
FIGURE 40 (A) FALSE-LOCK CAPABILITY PHASE DETECTOR AND ITS (B) TIMING DIAGRAM.....	45
FIGURE 41 CONVENTIONAL BINARY SEARCH BASED CONTROLLER .....	47
FIGURE 42 3-BIT UNBALANCE BINARY SEARCH ALGORITHM.....	48
FIGURE 43 COMPARISON WITH CONVENTIONAL BS AND UBS ALGORITHM .....	49
FIGURE 44 SIMULATION LOCK TIME VERSUS THE OPERATION RANGE.....	50
FIGURE 45 9-BIT STEP CONTROLLER .....	51
FIGURE 46 (A) 9-BIT BINARY CONTROLLER (B) SBG.....	52
FIGURE 47 THE OPERATION OF THE UBS CONTROLLER.....	53
FIGURE 48 THE ARCHITECTURE OF DCDL AND BWDC.....	54
FIGURE 49 THE SIMULATION RESULTS OF (A) DELAY TIME VERSUS INPUT VECTOR (B) POWER CONSUMPTION VERSUS DELAY TIME.....	54
FIGURE 50 LOCK PROCESS WHEN THE OPERATING FREQUENCY AT 100MHZ .....	55
FIGURE 51 LOCK PROCESS WHEN THE OPERATING FREQUENCY AT 125MHZ .....	56
FIGURE 52 LOCK PROCESS WHEN THE OPERATING FREQUENCY AT 250MHZ .....	56
FIGURE 53 LOCK PROCESS WHEN THE OPERATING FREQUENCY AT 500MHZ .....	56
FIGURE 54 THE PROPOSED ADSCM-DLL ARCHITECTURE.....	59
FIGURE 55 (A) THE BLOCK DIAGRAM OF PHASE DETECTOR (B) TSPC.....	60
FIGURE 56 THE OPERATION OF CONVENTION PD.....	61
FIGURE 57 THE MODIFY TSPC DFF .....	62
FIGURE 58 THE OPERATION OF PROPOSED PD.....	62
FIGURE 59 LADE.....	63
FIGURE 60 THE ARCHITECTURE OF PROPOSED DCDL.....	64
FIGURE 61 DEALT OF DCDL V.S. INPUT VECTOR.....	64
FIGURE 62 5-BIT LOCK-IN UNIT .....	65
FIGURE 63 THE OPERATION OF PROPOSED LOCK-IN UNIT .....	66
FIGURE 64 THE BLOCK DIAGRAM OF THE CALIBRATION UNIT.....	66
FIGURE 65 OPERATION OF DRPD [3].....	68
FIGURE 66 DRPD.....	69

FIGURE 67 THE PROPOSED INTERPOLATOR .....	70
FIGURE 68 THE LOCK DETECT UNIT .....	71
FIGURE 69 THE POWER COMPARISON OF WITH/WITHOUT LDU.....	71
FIGURE 70 LAYOUT VIEW OF THE ADSCM-DLL.....	72
FIGURE 71 LAYOUT VIEW OF THE TEST CHIP .....	72
FIGURE 72 THE OPERATION OF LOCK-IN STAGE.....	73
FIGURE 73 THE OPERATION OF CALIBRATION STAGE.....	74
FIGURE 74 THE PHASE ERROR OF EACH DELAY STAGE (A) 90NM (B) 130NM .....	75
FIGURE 75 ADSCM-DLL BASED DUAL CLOCK OUTPUT GENERATOR .....	78



# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND

With the growth of CMOS process technology, the complexity and operating frequency in the VLSI systems had growth exponentially. The design trend goes toward to the system-level integration and single-chip solution. In the point of System-On-Chip (SoC) design, the reusable modules takes advantages of design cycle and process portable. Therefore, the quality of the synchronous clock signals between each module becomes more important. How to eliminate the clock skew becomes an important issue for the high performance VLSI systems and SOC application.

Phase-locked loop (PLL) and delay-locked loop (DLL) are widely used to solve the clock synchronization problem. However, the DLL is more suitable for the clock de-skew problem than PLL due to the simple design effort and innate characteristic. Besides, the DLL also provides better jitter performance because there is no jitter accumulation in a voltage controlled delay line (VCDL) or digitally controlled delay line (DCDL). As a consequence, the DLL is frequently used in clock synchronous.

### 1.2 MOTIVATION

The application of DLL is not only limited to the clock synchronous but also for the clock/data recovery (CDR) circuit [45], double data rate (DDR) SDRAM [9], [10] and frequency multiplier [43], [44], [1]. A multiphase VCDL or DCDL output is typically used to implement this circuit function. However, the edges of the multiphase output signals are not equally spaced due to the delay mismatches. For the CDR circuits using the multiphase sampling schemes [45], the phase offset corrupts signal constellation and raising the bit error rates. Similarly, the frequency multiplier

using edge combiner schemes [43], [44], the static phase error among each delay stage induce the fixed pattern jitter at the multiplied clock output. Therefore, a DLL with precise multiphase outputs is necessary. Moreover, the conventional DLLs may suffer from harmonic lock over a wide operating frequency range. Various wide-range DLLs architectures have been develop to solve the false locking problem. The DLL with multiple VCDLs to overcome this problem of a limited delay range is proposed in [14]. In [6], an all-analog DLL improves the locking range by using replica delay line. However, it is not suitable for the process portability and noise immunity consideration. Therefore, digital is developed to improve this problem.

According to above issues, this thesis focuses on the techniques of the search algorithm for the DLL to eliminate false locking problem and the calibration mechanism for the multiphase outputs to compensate the delay mismatch among the delay line.

### 1.3 ORGANIZATION

The thesis organization is as follows:

Chapter 2 gives an overview of DLL, including analog DLL and digital DLL. A comparison result is also given in this chapter.

Chapter 3 describes the fundamentals of the calibration schemes for multiphase DLL and presents a novel rapid self-calibrated (RSC) algorithm. Base on the RSC algorithm, the multiphase DLL can adjust the phase difference in digital manner and eliminate the phase error of the multiple outputs. A comparison result with other calibration schemes is also presented.

Chapter 4 A modify binary search algorithm is presented which extends the locking range to fully delay line and avoid harmonic locking, simultaneously. Then, the circuit design and detail operation flow of the DLL which base on modify binary search algorithm will also be addressed.

Chap 5 presents a multiphase DLL with precise multiple outputs. Base on the RSC algorithm, the system architecture and its circuit design is also presented. Finally, we will show the implementation of layout, simulation result and performance summary.

Chap 6 presents the conclusion and future work.



# CHAPTER 2

## AN OVERVIEW OF DELAY-LOCKED LOOP

### LOOP

### 2.1 BASIC CONCEPTS OF DELAY-LOCKED LOOP

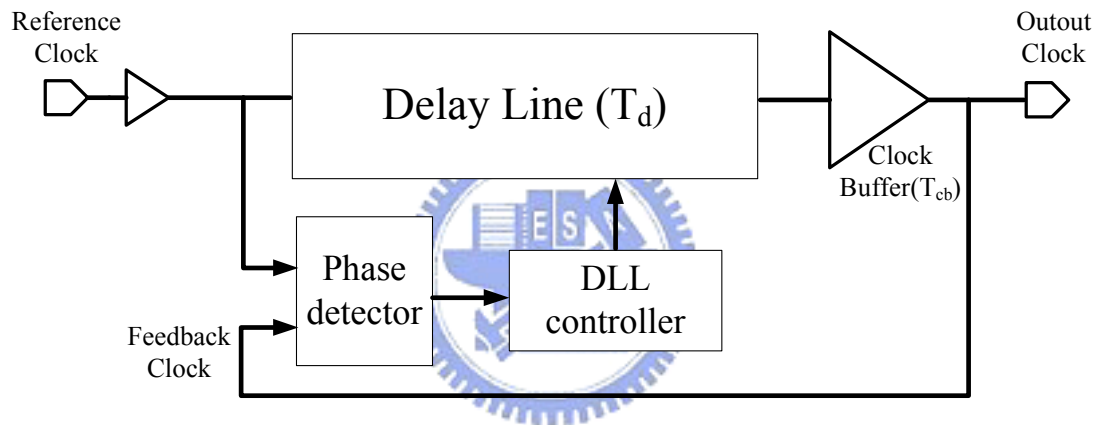


Figure 1 The architecture of conventional DLL

The basic architecture of a conventional DLL is shown in Figure. 1. A DLL consists of a phase detector, a variable delay line, and a DLL controller to convert the PD's output signal to digital or analog signals for the delay line. It automatically tunes the delay time of the delay line and inserts an optimal delay time ( $T_d$ ) to compensate the phase error between the reference clock and output clock. After the DLL is locked, equation (2.1) will be satisfied, where  $K$  is an integer,  $T_{ref}$  represents the clock period of the reference clock.  $T_d$  and  $T_{cb}$  denotes the delay time of delay line and clock buffer respectively.

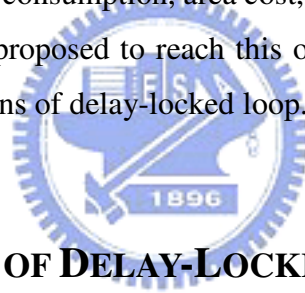
$$K T_{ref} = T_d + T_{cb} \quad (2.1)$$

According to equation (2.1), when the DLL is locked, there is no phase error

between the reference clock and output clock (or called feedback clock). At the same time, the output clock will be synchronized with the reference clock, and the clock buffer delay can be ignored.

Since the delay line is adjusted in the analog manner, the continuous tuning step results in higher delay resolution than in a digital one. Besides, it can achieve better jitter performance and smaller chip area. However, it is not suitable for future low-voltage applications because it cannot provide enough delay range under low supply voltage [4]. Moreover, the process-sensitive characteristic makes them difficult to be transferred to advanced technologies and less noise immunity in a System on Chip (SoC) environment. On the contrary, digital manner can provide more robust to overcome the process, voltage, and temperature (PVT) variations, and exhibit shorter lock time and noise immunity than analog one.

The design challenge of DLL is how to overcome the PVT variations, and balance the clock jitter, power consumption, area cost, portability, and lock time. Thus, different manners have been proposed to reach this objective. In the next section we will introduce the classifications of delay-locked loop.



## **2.2 CLASSIFICATIONS OF DELAY-LOCKED LOOP**

We can classify delay-locked loop into open loop type and closed loop type by different locking mechanisms.

### **I. Open loop type Delay-Locked Loop**

Synchronous mirror delay (SMD) is the most typical circuit of open loop type design. The main advantage of SMD is the fast locking characteristics in recovering from power-down or standby mode within a few cycles of the system clock. Nevertheless, the fast locking characteristics of SMD, the phase error between the reference clock signal and output signal cannot be controlled as accurately as a close loop type DLL [46]. Thus the analog synchronous mirror delay (ASMD) [49] was proposed to enhance the phase acquisition performance.

### **II. Closed loop type Delay-Locked Loop**

Register-controlled DLL [15], [29] and Counter-controlled DLL [48] is the most typical example of closed loop type delay-locked loop design. The most advantage is the improvement of the clock skew problem in open loop type delay-locked loop caused by environment variation, and smaller static phase error and lower clock jitter is also achieved. However, in order to synchronize between the reference clock signal and output clock signal the lock time in closed loop type is longer than in an open loop one and the lock mechanism also consumes more power. Thus, the Successive Approximation Register-controlled DLL [4] that uses binary search manner and Time measurement controlled DLL has been proposed to resolve lock time and power consumption problem.

Besides the classifications mentioned above, we can define different types of DLLs by circuit implement manners. The classifications of DLL circuit are defined as follows :

(1) Analog DLL : Each block processes an analog signal. The advantages are low jitter output and higher delay resolution. The disadvantage is lower noise immunity and a longer design cycle.

(2) All Digital DLL : Each block processes digital signal. Higher noise immunity and portability are the advantages of ADDLL. However, the lower delay resolution and jitter performance are disadvantages in ADDLL in general.

(3) Mixed DLL : Using digital blocks to reach fast coarse tuning lock and fine tuning the phase error in an analog manner. The advantage is that it can reach high delay resolution and fast lock time, but the drawback is it is hard to integrate digital and analog blocks simultaneously.



## 2.3 DESIGN OF ANALOG DELAY-LOCKED LOOP

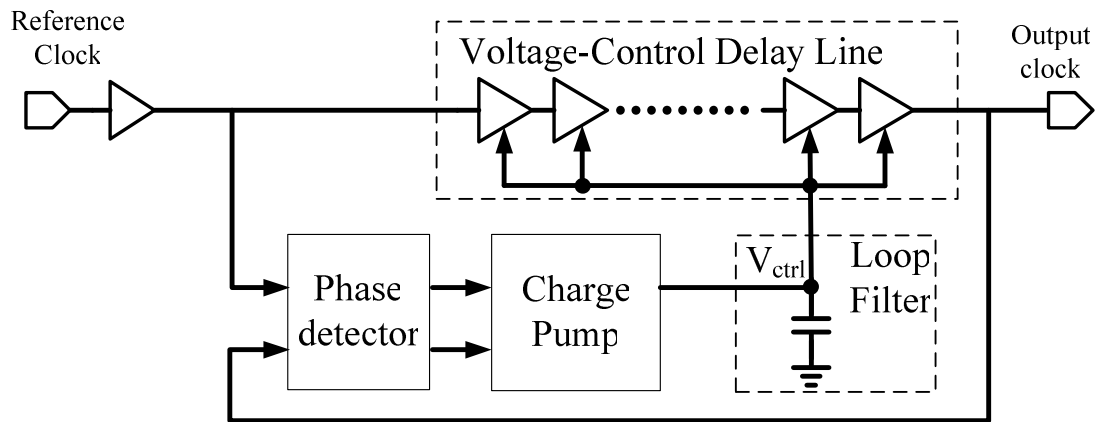


Figure 2 Block diagram of analog DLL

Figure 2 illustrates the block diagram of an analog DLL that contains a voltage-controlled delay line (VCDL), a phase detector, a charge pump, and a first order loop filter. The reference clock signal propagates through the voltage-controlled delay line that consists of cascaded variable delay stages. The phase detector compares the phase between the reference clock and output clock, which is the delay version of VCDL, and produces an up/down signal. The charge pump integrates the phase detector output signal and the loop filter produces a control voltage,  $V_{ctrl}$ , to operate the delay line.

### 2.3.1 Voltage-controlled Delay Line

Delay elements are widely used in digital systems and are essential parts for clocking operation in high speed VLSI application. The simple and easy to design makes the RC delay and inverter chain method have been the most common delay elements in those applications. However, the characteristics of the delay element are sensitive to supply noise and PVT variations.

In this section, we will introduce the two distinct approaches of VCDL. They are RC-time-constant Controlled Delay Line (RCCDL) and Current-Starved Controlled Delay Line (CSCDL)

## 1. RC-time-constant Controlled Delay Line

The basic delay line of RC-time-constant controlled delay element is shown in Figure 3(b). The circuit can be obtained by cascading even number of the same delay elements. In Figure 3(a), the control voltage ( $V_{ctrl}$ ) controls the charge current. The transistor Mn1 in essence controls the amount of effective load capacitance “seen” by the driving gate. Large value of  $V_{ctrl}$  decreases the resistance of the transistor Mn1, so the effective capacitance at the logic gate output increase, producing a large delay.

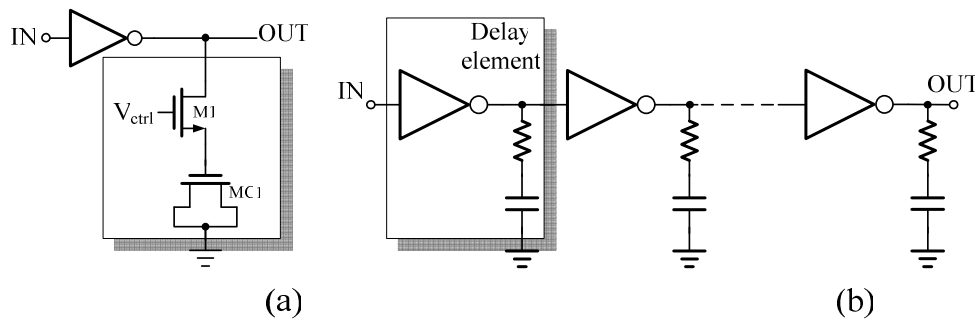


Figure 3 The RCCDL (a) delay element (b) delay line

## 2. Current-Starved Controlled Delay Line

A basic delay element of CSCDL is shown in Figure 4(a). A simple current mirror can be used to generate two bias voltages. The control voltage  $V_{ctrl}$  is applied to a series-connected element which can “current starve” an inverter.  $V_{ctrl}$  modulates the ON resistance of pull-down transistor Mn1, and through a current mirror, pull-up transistor Mp1. These variable resistances control the current available to charge or discharge the load capacitance. Large values of  $V_{ctrl}$  allow a large current to follow, producing a small delay.

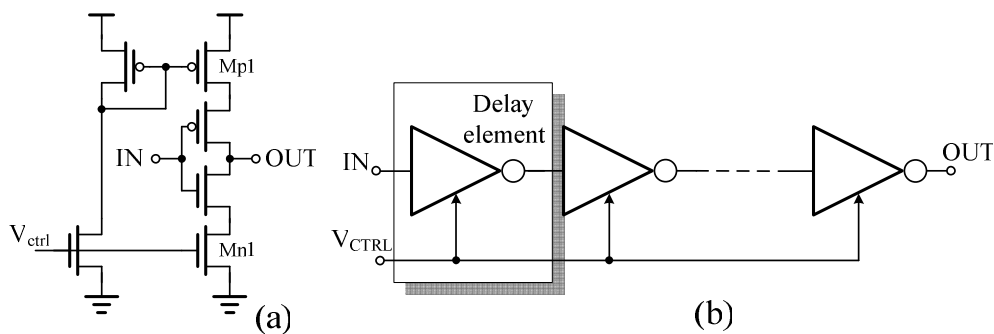


Figure 4 The CSCDL (a) delay element (b) delay line

## 2.3.2 Phase Detector

Phase Detector is a circuit that is response the relationship between reference and feedback signal. Figure 5 shows three-state phase detector circuit and Figure 6 shows the waveforms in some conditions. Unlike multipliers and XOR gate, three-state PD generates two outputs that are not complementary. When the feedback signal is high and the reference signal is low, then the PD produces positive pulse at down signal, while up signal remains at zero.

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

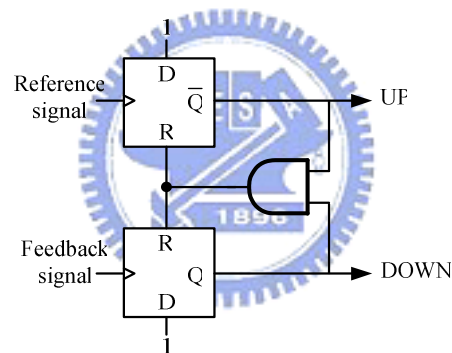


Figure 5 Three-state phase detector

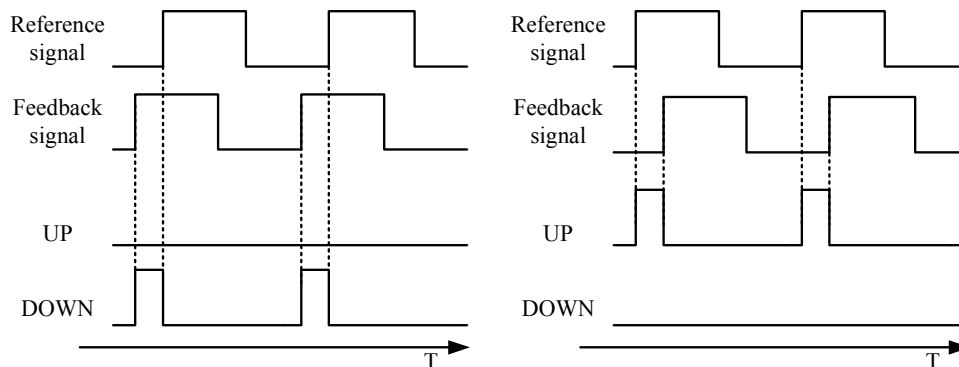


Figure 6 PD responses with (a) reference signal lag feedback signal (b) reference signal lead feedback

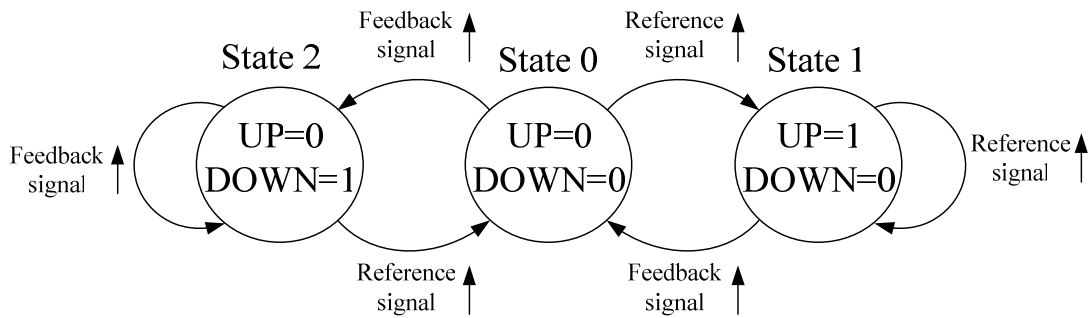


Figure 7 PD state diagram

In the Figure 7, it shows the PD circuit behavior. It has three state diagrams: UP=1, DOWN=0 (state 1), UP=0, DOWN=0 (state 0), UP=0, DOWN=1 (state 2). Because the PD is build up from two edge-triggered sequential circuits, we can avoid dependence of the output upon the duty cycle of the inputs.

Suppose the circuit is initially in state 0. Then a rising edge on reference signal takes the circuit to state 1, where UP=1, down=0. With state 1 is reached, any more rising edges at reference signal won't case state change at all. The circuit will remain in this state until a transition occurs on feedback signal, upon which the PD returns to state 0. The switching sequence between state 0 and state 2 is similar.

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

### 2.3.3 Charge Pump and Loop Filter

The simple model of charge pump and loop filter is shown in Figure 8. It consists of two matched current sources and function as switch. When the up signal is high, it turns on the upper switch and charges output node Vctrl. On the other hand, when the down signal is high, the down signal turns on the lower switch and discharges the output node Vctrl. Finally, if both up and down signal are low, then net current is zero and output node Vctrl holds the original voltage.

The loop filter can be either passive or active. In general, a passive filter is

simple to design and has better noise performance. The passive filter was shown in Figure 9, which may be first-order, second-order, or other high order structure. High order filters take advantages of rejecting out-band noise. However, low order filters result in more stable operations. The choice between high order filters and low order filters depends on the applications and to prevent DLL into unstable state.

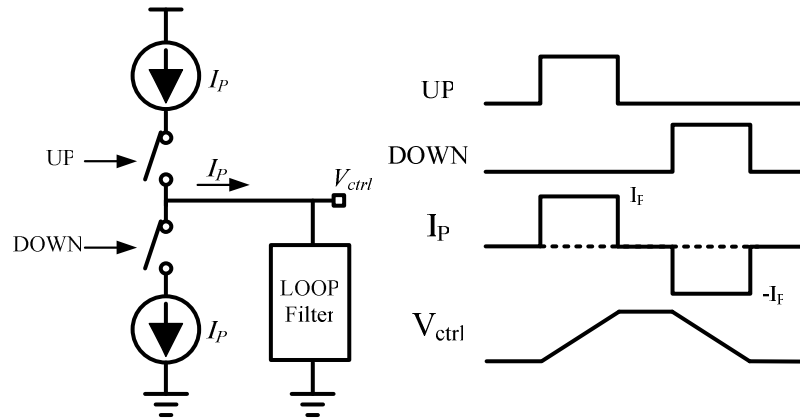


Figure 8 Simple model of charge pump and loop filter

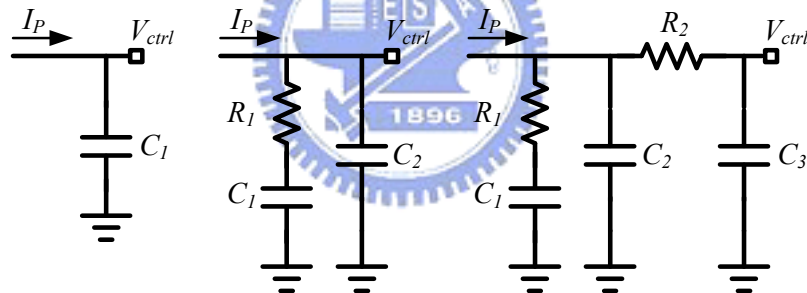


Figure 9 Loop filter

### 2.3.4 Stability Analysis of Delay-Locked Loop

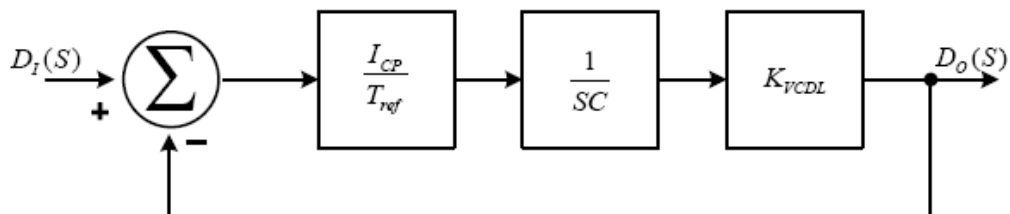


Figure 10 Small signal AC model of the conventional analog DLL.

Before starting the stability analysis of ADLL, the small signal AC model shall be introduced first. This is shown in Figure 10 where summer stands for phase detector,  $I_{CP}$  is the charge pump current,  $T_{REF}$  is the period of input reference clock,  $C$  is the capacitor value in loop filter, and  $K_{VCDL}$  is the gain of VCDL. When loop is in steady-state locked condition, the s-domain transfer function from input to output is

$$\frac{D_0(s)}{D_1(s)} = \frac{1}{1 + \frac{s}{W_N}} \quad (2.2)$$

Where

$$W_N = \frac{2p}{T_{REF}} \quad (2.3)$$

From Eq. 2-10, we can easily find that the DLL is a first order system that is inherently stable. Unlike the small-signal AC model for a typical PLL, a minimum of a second order transfer function is required.

Since the transfer function is inherently stable, a wider loop bandwidth can be used. This allows a fast acquisition time, as well as the use of small loop filter capacitors facilitating integration. However, the small-signal AC model is only valid when the loop bandwidth, that is  $\omega_N$ , is much smaller than the phase detector comparison frequency (generally 10:1). Therefore, the following equation should be satisfied for stability consideration.

$$\frac{I_{CP} \times K_{VCDL}}{2p C} \frac{1}{W_{REF}} \frac{1}{10} \quad (2.4)$$

Where

$$W_N = \frac{2p}{T_{REF}} \quad (2.5)$$

## 2.4 DESIGN OF CONVENTIONAL DIGITAL DLL

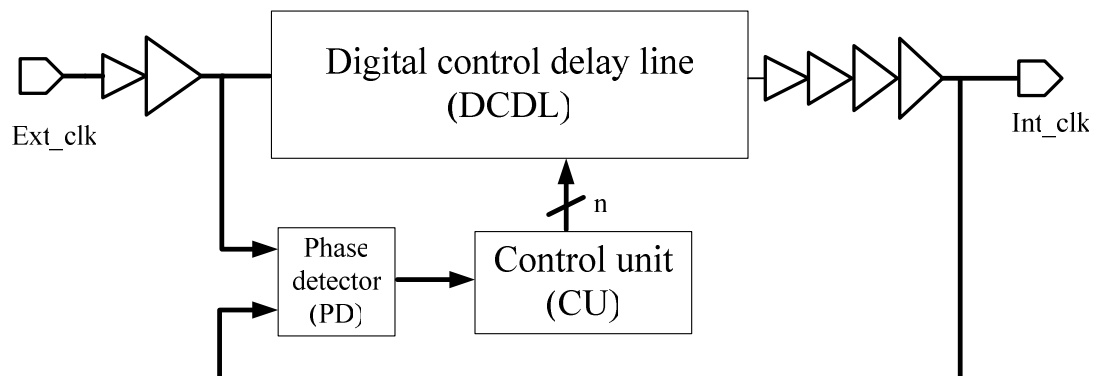


Figure 11 Block diagram of digital DLL

The conventional digital DLL architecture is shown in Figure 11. It consists of three major blocks and constructs a close loop circuit. The major blocks are phase detector (PD), control unit (CU) and digital control delay line (DCDL) respectively.

The input of DLL is external clock (Ext\_clk) and feedback signal is internal clock (Int\_clk) which is the delayed version of the external clock signal. The CU generates digital signals to control the amount of the delay time, and the PD detects the phase error between the input signal clock signal and the feedback signal. If Ext\_clk signal leads Int\_clk signal, the CU adjusts the digital signals to increase the delay time of DCDL. Conversely, the CU decrease the delay time to compensate the phase error until the Int\_clk synchronize to Ext\_clk.

By different implementation of control unit, we can classify control unit into register-controlled, counter-controlled, successive approximation register-controlled, and time measurement controlled of conventional DLL. The following section will describe in detail.

### 2.4.1 Register-controlled DLL [15]

As Figure 12 shows the block diagram of register-controlled DLL. The n-bit shift register which is controlled by the output of phase detector is used to generate control signals for the digitally controlled delay line. At any time, only one bit of the shift

register is active to select a specify delay time of delay line. The phase detector detects the relation between input clock and output clock, and generates left and right signal for shift register to control the amount delay time. When Enable is active, it will enable the shift register, vice versa.

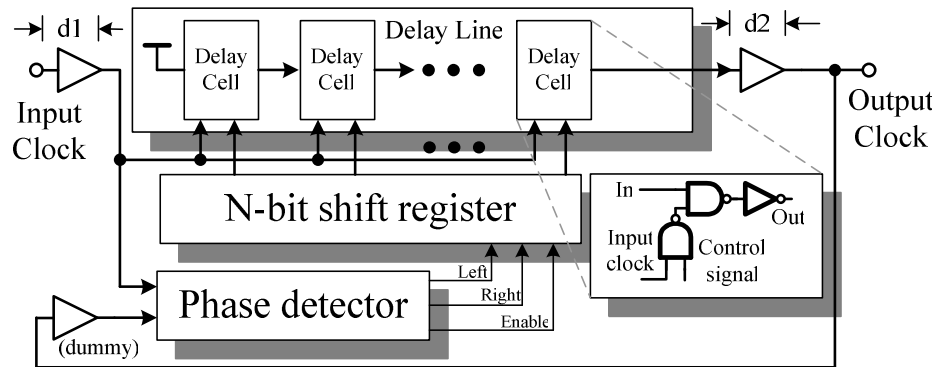


Figure 12 Register controlled DLL

When the output clock leads the input clock, the phase detector sends left signal to shift register and the high bit in the shift register will be shifted left to increase the delay time to compensate for the delay mismatch. Similarly, when the right is active, the high bit in the shift register will be shifted right to decrease the delay time. When Enable is active, the phase error between the input clock and the output clock is within one unit delay, and the data in the shift register will be held. Under this mechanism, the loop is locked and the phase error will not exceed the unit delay.

Although the control mechanism is quite sample, but when the operating range is increased, the additional delay stages of delay line should be added, however, it increases the chip area. Beside, the control mechanism is one by one, means, the more delay stages needs more shift registers to control the delay line. Thus, it also increases locking time. In the worst case, n-bit shift register needs  $n/2$  locking cycles.

## 2.4.2 Counter-controlled DLL [48]

Basically, the operating principle of counter-controlled DLL is similar to register-controlled DLL expect the up/down counter substitutes for the shift register to control the delay line. In addition, the binary-weighted delay line is adopted and no longer consists of delay stages with equal delay time. The linearity of binary-weighted



delay line is an important issue, we will discuss in section 2.4.5. Hence, we focus on the characteristic of CDLL.

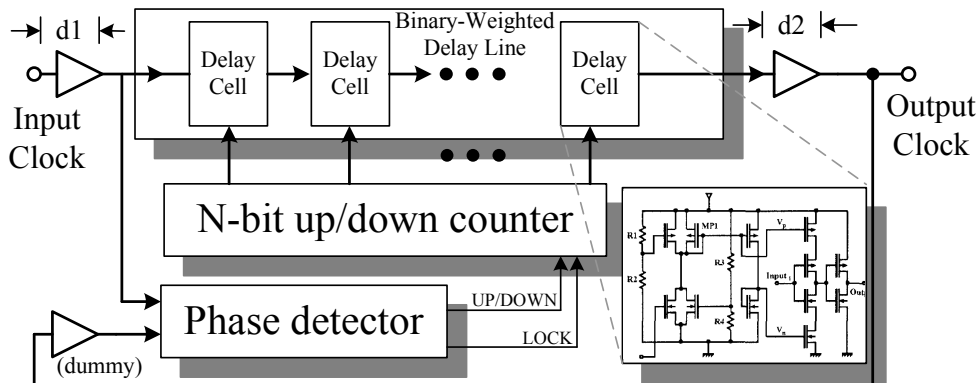


Figure 13 counter-controlled DLL

Figure 13 shows the block diagram of counter-controlled DLL. The active of up/down counter is base on the output of phase detector. The n-bit control word determiners whether the input signal goes through the delay path or passes it. The most different between register-controlled DLL (RDLL) and counter-controlled DLL (CDLL) is area requirement. For example, compare with the RDLL, if 128 delay stages are required in a RDLL, only 7 delay stages are required in a CDLL. Besides, the 128-bit shift register in a RDLL can be substituted for 7-bit up/down counter. While the operating ranges and delay resolution of RDLL and CDLL are the same, the delay line of RDLL will get larger offset delay time and occupy larger chip area than the CDLL. By using CDLL, the chip area could be reduced while maintaining the same operating range as in a RDLL. However, the CDLL still use to linear approach manner to trace the input clock, thus the locking time of CDLL would not get any improvement as RDLL. In the worst case, with n-bit binary-weighted delay line, the locking time maintains  $n/2$  locking cycles.

### 2.4.3 Successive Approximation Register-controlled DLL [4]

As we mention above, the locking time is an important parameter for digital DLL to evaluate the performance, especially in the high-speed memory applications. Both of the DLL that mentioned above based on the linear search exhibit the same lock time. The linear search algorithm increases the locking time when finding the optimal delay

of delay line to insert into the input clock and output clock. The binary search algorithm may be applied to reduce the locking time. First, the most significant bit (MSB) of the control word is set to 1, and the other bits all are set to 0. The phase detector judge whether the output clock leads the input clock or not. If output clock leads the input clock, the MSB is set to low. If output clock lags the input clock, the MSB remains high and held constant. In this way, the MSB is determined. The operating produce is repeated for the following bit until the least significant bit (LSB) is determined.

Figure 14 shows an example of the 3-bit binary search algorithm. Assume the final control word is set to “001” and the initial control word is set to “100”. In this example, the output clock leads input clock in the step 1 and step2, and output clock lags input clock in the step 3. Finally the binary searching finds the correct control word “001”.

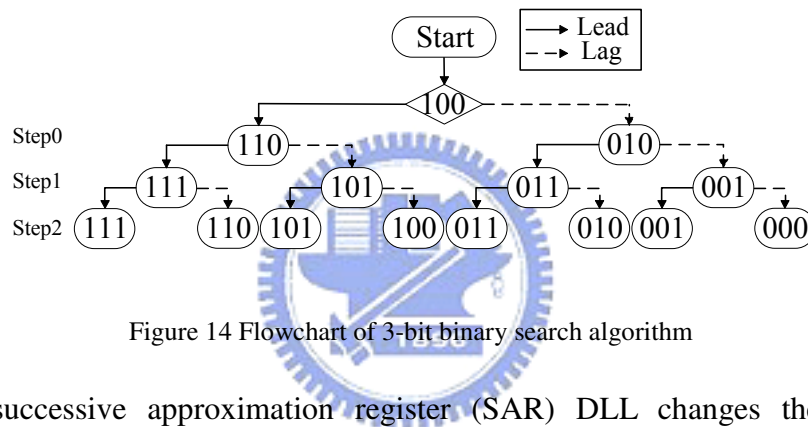


Figure 14 Flowchart of 3-bit binary search algorithm

The successive approximation register (SAR) DLL changes the searching mechanism to binary search algorithm and adopted with binary-weighted delay line. It is not only reduces the chip area but also shorten the locking time. In the worst case, with n-bit delay line, the locking time of SAR-DLL is  $\log_2(2^{(n-1)})$ . Unfortunately, The SAR controller in the DLL determines the value of each bit of the word in a sequential and irreversible. Therefore, it becomes an open-loop type circuit after lock-in and never against the PVT variation. An improved SAR DLL [41] was proposed to solve this problem by using the counter-controlled control word instead of SAR-controlled. The initial control word of the counter is load from the SAR controller, and then a counter-controlled DLL is started to maintain the environment variation.

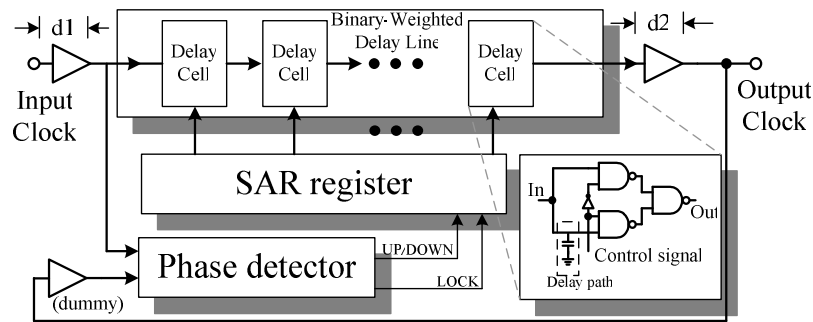


Figure 15 SARDLL

### 2.4.4 Time measurement controlled DLL

Another mechanism to reduce the locking time was proposed in [49] [32]. The time measurement controlled (TM) DLL divide the locking produce into two stages, coarse tuning and phase tracing. The coarse tuning stage is based on the time to digital converter (TDC) circuit. In RDLL and CDLL, the narrow tuning step causes the long locking time. The TDC can measures the input clock period and convert it to digital signals within two clock cycles, then transfer the digital control word to the control block, therefore, the tuning step is extensive. After the coarse tuning stage, the phase tracing stage is active to fine tune the delay of the delay line. Usually, only few control bits need to be determined in the phase tracing stage, therefore, a counter-controlled based control block is preferred. Compare with the TD-DLL and SAR-DLL, there is no different of locking time in phase tracing stage, the most distinction between TD-DLL and SAR-DLL is in the coarse tuning stage. The locking time in the coarse tuning stage of SAR-DLL depends on how many control bits need to be determined, but the TD-DLL can achieve coarse truing within only few cycles. In the worst case, assume  $m$  fine tuning bits, the locking time of TD-DLL is  $(m/2+2)$  locking cycles. Although the search time of TD-DLL is quite quick, the drawback of TD-DLL still is the area requirement.

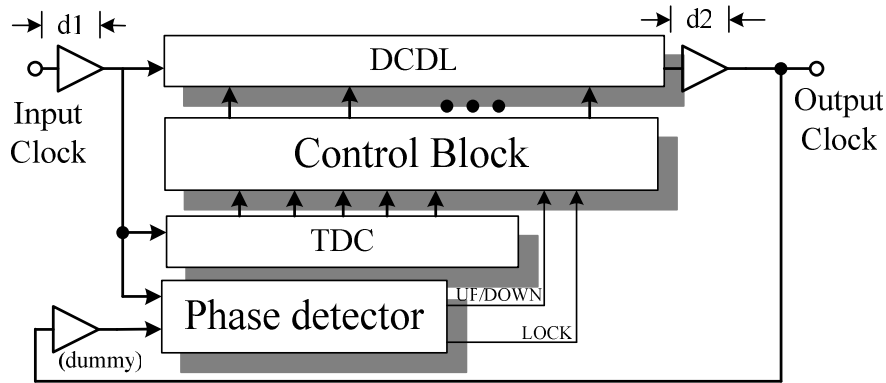


Figure 16 TM DLL

## 2.4.5 Digitally Controlled Delay Line

Digitally controlled delay line (DCDL) is the key component of ADDLL. Like most voltage controlled delay line (VCDL), the DCDL consists of several different digitally controlled delay elements (DCDE). There are two main parameters to adjust the delay time of DCDL. One is the total number of the delay elements, usually taken for the coarse tune method, and the other is the propagation delay time of the delay elements (i.e. inverters), which is usually taken for the fine tune method. The first delay time adjustment parameter is usually realized by a path-selection approach, and Figure 17 shows the example [16]. In this example,  $2^n$  delay buffer are connected in series. A decoder decodes an  $n$ -bit control word  $D$  into  $2^n$  control lines. Hence, if the propagation delay time of each buffer stage is  $T_{\text{buffer}}$ , then the time resolution is  $2 * T_{\text{buffer}}$ .

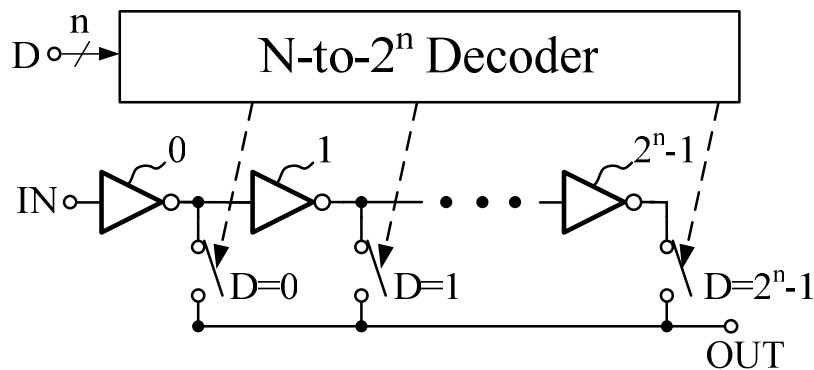


Figure 17 DCDL realized by a path-selection method

Another example of phase-selection method is shown in Figure 18. The lattice delay line (LDL) [5] cascaded several lattice delay units (LDU). The digital control word  $T$  determines the clock signal ( $CLK_{IN}$ ) propagation path. Unlike conventional digital controlled delay element with two different delays controlled by a multiplexer increasing tuning range but intrinsic delay increases as well. When the tuning range increases, the minimal delay is not changed. Both the intrinsic delay and the delay step in an LDL are the delay of two NAND gates. As the operating frequency increases, the number of activated delay units is reduced and the power consumption remains the same.

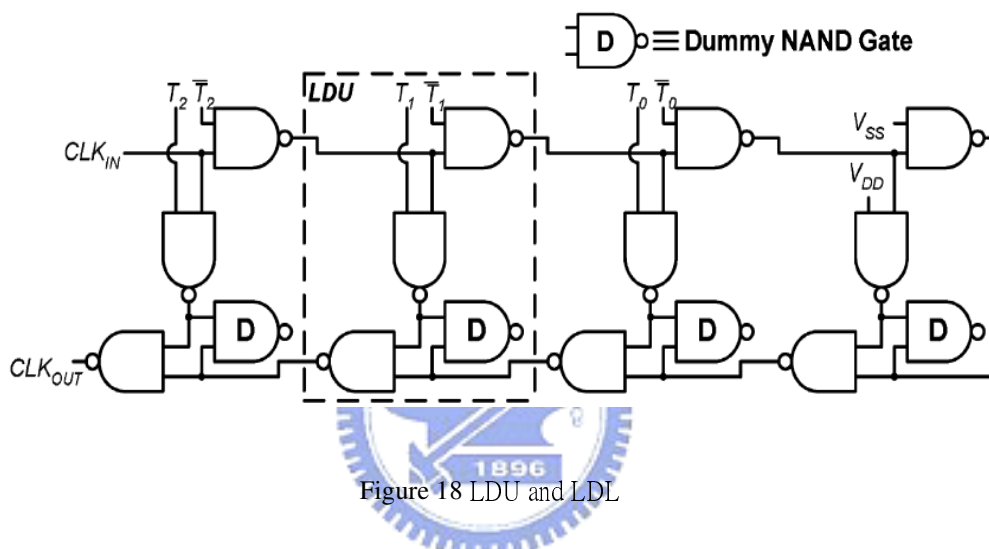


Figure 18 LDU and LDL

There are several different architectures that have been used to implement a DCDE. However they can generally be classified into the shunt capacitor based, the parallel-inverter based and the current-starved based delay elements. In the following section, we will introduce different kinds of DCDE.

### 2.4.5.1 Shunt Capacitor based DCDE

Figure 19 shows the basic circuit of using a shunt capacitor based DCDE [50]. In this circuit,  $MC1 \sim MCn$  acts as shunt capacitor. Transistor  $M1 \sim Mn$  controls the charging and discharging current to the  $MC1 \sim MCn$ . The operating is similar to RCCDL; replace the  $V_{ctrl}$  to the digital control word  $D$  which is  $n$ -bit resolution controls the equivalent capacitance on the output node. As a consequence, the delay time of shunt capacitor based method can be controlled in binary-weight. The drawback of shunt capacitor based DCDE is sensitive to power supply noise and PVT variation.

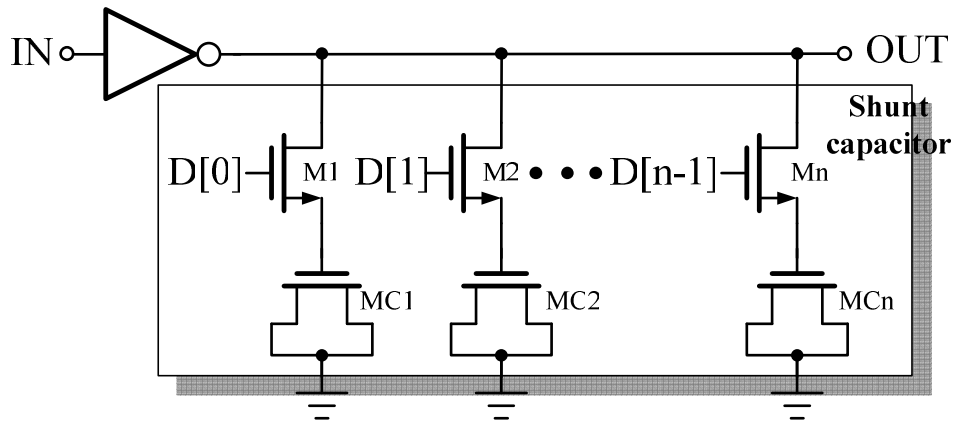


Figure 19 shunt-capacitor based DCDE

### 2.4.5.2 Standard Cell based DCDE

One simple example of standard cell based DCDE was proposed in [20] [51], as shown in Figure 20. The delay element is cascaded six inverters in the first row and the additional tri-state inverter with its control bit is added in every column. By enabling the number of tri-state inverter buffer, the delay time of DCDE can be controlled. It is simple and easy to implement. However, it needs large area and high power dissipation for the fine tune necessarily in the DCDL design. Besides, the resolution is hard to be uniform.

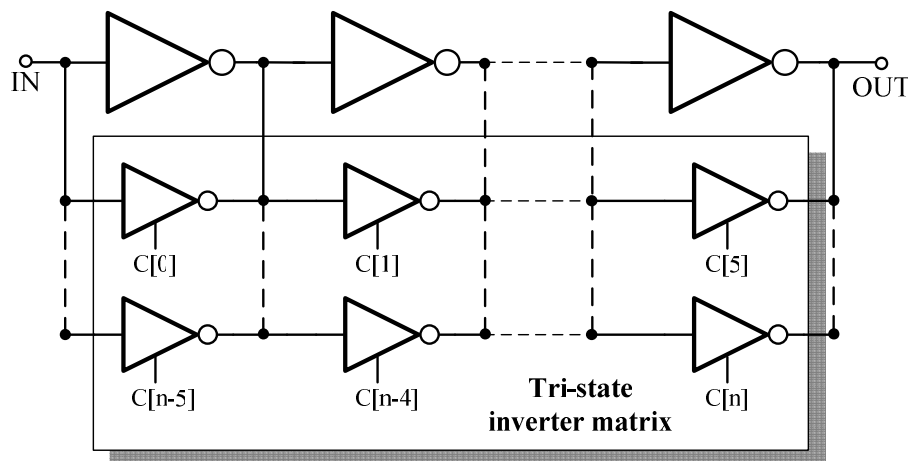


Figure 20 parallel tri-state inverter based DCDE

The other example, as shown in Figure 21, the DCDE is implemented by an add-or-inverter (AOI) cell and or-and-inverter (OAI) cell with two parallel tri-state

inverters was proposed in [48]. The basic method is to adjust the driving capability with resistance control. The advantage is that this fine tune method of DCDE has less area and power dissipation compare with [20] [51]]. However, since it's based in AOI-OAI cell to change the delay resolution, the resolution step is also hard to be uniform and sensitive to power-supply variation. Besides, it also requires an additional decoder for mapping the control input of AOI-OAI cell.

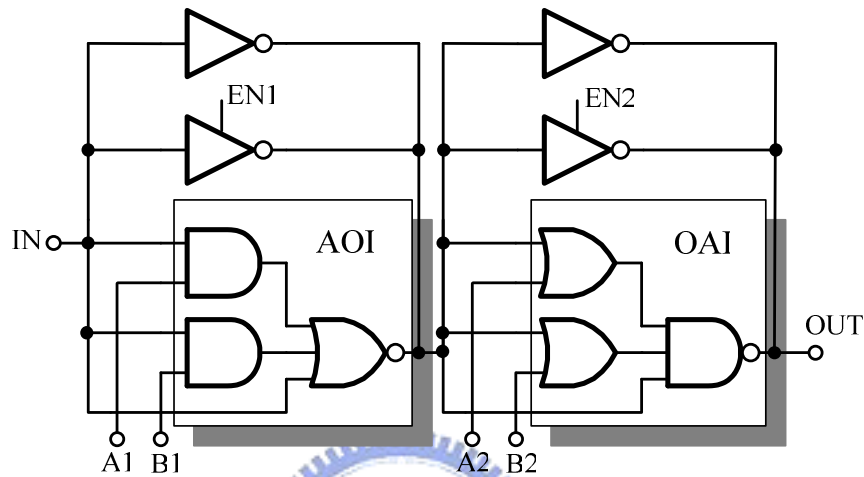


Figure 21 AOI-OAI parallel based DCDE

### 2.4.5.3 Current-starved based DCDE

The current starved based DCDE was proposed in [26]. As Figure 22 shows, the charging and discharging currents of the inverter, composed of M1 and M2, are controlled by two sets of current-controlling nMOS ( $M_{n0}, M_{n1}, \dots$ ) and pMOS ( $M_{p1}, M_{p2}, \dots$ ) transistors at the source of M1 and M2, respectively. The current controlling transistors are sized in a binary fashion. It allows achieving binary incremental delays. As can be seen, by applying a specific binary vector to the controlling transistors, a combination of transistors is turned on at the sources of M1 and M2 transistors. Such an arrangement controls the rise time and fall time of the output voltage of the inverter.

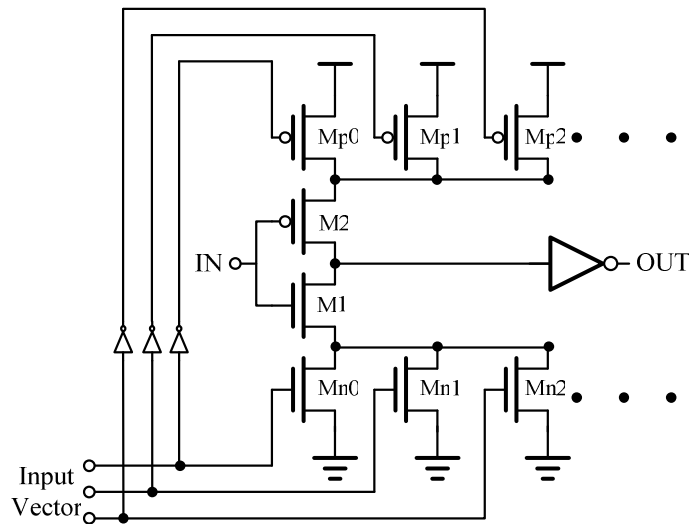


Figure 22 current starved based DCDE

However, one of the problems with the current starved based DCDE architectures is the non-monotonic delay behavior with ascending binary input vector. As can be seen in the circuits of Figure 23, the input vector changes the effective resistance of transistors placed at the source of the nMOS or pMOS transistors of the inverter. This not only changes the resistance at the source of M1 or M2, but also changes the parasitic capacitance associated with transistors at these nodes. This is because the parasitic capacitance at the drain of a MOSFET is different in the ON and OFF states.

In [8], there are two factors depending on the input vector to affect the delay :

(1) The resistance of the controlling transistors:

The circuit delay can be increased / decreased by increasing / decreasing the effective ON resistance of the controlling transistors at the source of M1.

(2) The capacitance of the controlling transistors:

The charge sharing effect cause the output capacitance to be discharge faster and the overall delay decrease as the effective capacitance of the controlling transistors at the source of M1 increase.

The larger resistance increases the delay; however, larger parasitic capacitance decreases the delay. The effective capacitance seen at the source of M1 depends on which controlling transistors are on. Because of the ON and OFF capacitances between drain and ground of a MOSFET is different. Therefore, it may make monotonic



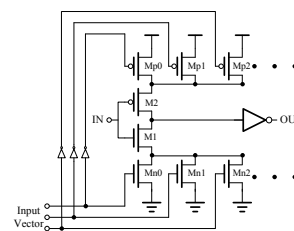
characteristic of the DCDE can not be ensured with ascending input vector. This situation will be further complicated as the number of delay controlling transistors increases. Table 1 shows the comparison of the different type of DCDE.

Table 1 Comparison of different type DCDE

Delay cell type	Drawbacks	Circuit structure
Shunt-capacitor based	<ul style="list-style-type: none"> <li>➤ Sensitive to power supply noise</li> <li>➤ Process mismatch</li> </ul>	
Standard-cell based	<ul style="list-style-type: none"> <li>➤ Larger area and power dissipation</li> <li>➤ Delay resolution</li> <li>➤ Different coarse delay and fine delay cells</li> </ul>	

Current-starved based

- Poor linearity
- Sensitive to PVT variation



## 2.5 COMPARISON OF DIGITAL DLL AND ANALOG DLL

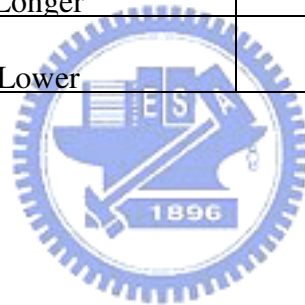
The most advantage of the analog approaches is the smaller static phase, good jitter performance, fine resolution because the delay is varied continuously. In addition, the analog DLL achieves small chip area and low power consumption. However, it suffers from slow locking and performance degradation due to sensitivity to variations of process and temperature. Although digital requires more chip area and power

dissipation, it is more robust against process, voltage, temperature (PVT) variation. Besides, the digital DLL provides fast lock time and easy to design. However, the quantization error of the digital DLL is unavoidable because the delay adjustment is in a discrete manner.

However, the digital DLL is still attractive of its shorter lock time and easy integration compare with analog approach. Table 2 shows the comparison of the analog DLL and digital DLL.

Table 2 Comparison of analog DLL and digital DLL

	<b>Analog</b>	<b>Digital</b>
Phase error	Smaller	Larger
Lock range	Smaller	Larger
Lock time	Longer	Short
Noise	Lower	Higher



# CHAPTER 3

## MULTIPHASE DELAY-LOCKED LOOP WITH SELF-CALIBRATION

In this chapter, it introduces the multiphase DLL with self-calibration. The conventional multiphase DLL architecture, design consideration and self-calibration schemes would be described in Section 3.1 and Section 3.2, respectively. In addition, the applications of multiphase DLL would be detailed in Section 3.3. Finally, Section 3.4 would give an introduction of proposed Rapid Self-Calibration (RSC) algorithm.

### 3.1 INTRODUCTION OF MULTIPHASE DLL

A ring-oscillator-based phase locked loop (PLL) or delay line based delay-locked loop (DLL) has been widely used because of its ability to generate multiphase clock signals. The multiphase clock signals can be used in various applications. Time-interleaved architectures, like a transmitter and receiver, employ multiple signals processing paths in parallel to achieve high overall speed while the speed of each channel is standard [48] [24]. In wireless communication systems, the multiphase clock signals are easily converted into the in-phase and quadrature (I/Q) signals with  $\pi/4$  radian difference essential for the down-conversion mixer [25]. In frequency synthesizer, the multiphase signals are used to generate a high frequency signal [1].

In most of these systems, the ring oscillator or delay line which consists of several identical delay elements is inserted into a negative feedback loop. When the PLL or DLL into the locked state, that means, the reference clock signal is split in several identical parts and the delay time of each delay stage is equally. Unfortunately, even all the delay stages are designed to be identical, each delay stage introduces a different delay due to the mismatch after fabrication, not to mention temperature and supply

voltage variations.

### 3.2 APPLICATION OF MULTIPHASE DLL

In this section, we will introduce the application of multiphase DLL in detail. In these applications the multiphase DLL is used to replace the PLL. The choice of DLL rather than PLL is due to the fact that they do not exhibit the jitter accumulation characteristic and there is no need for frequency multiplication of some applications.

#### I. Frequency Synthesizer

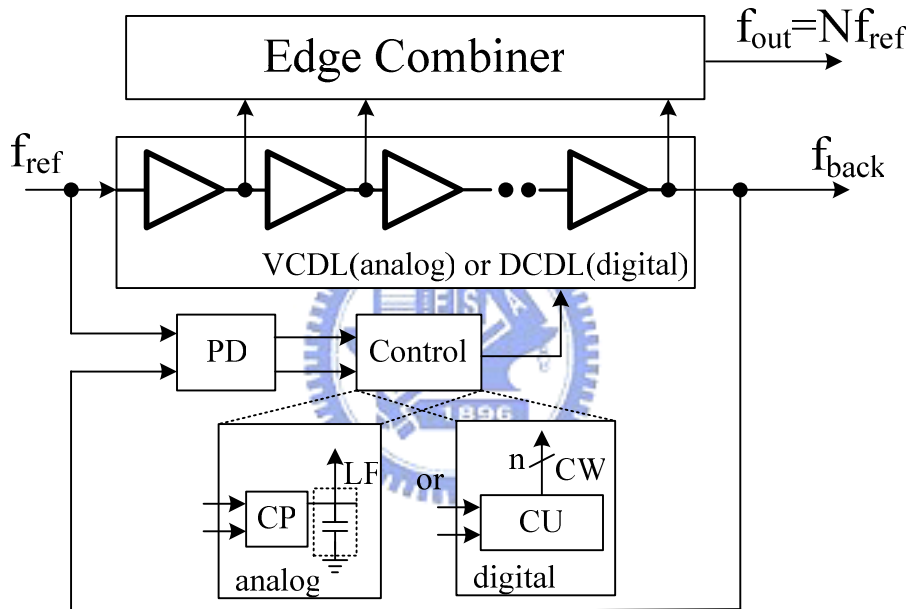


Figure 23 The block diagram of conventional DLL-based frequency synthesizer.

A DLL can operate as PLL, which uses delay line to replace VCO. Fig. 23 shows the simplified block diagram of DLL-based frequency synthesizer. When the loop is locked, the output phases of every delay stage are evenly spaced one reference clock period  $T_{ref}$ . Each phase difference of two delay stage has a delay of  $T_{ref}/N$  and the edge combiner can generate a transition for each phase output transition, hence the output frequency is the  $N$  times the reference frequency  $T_{ref}$ .

A multiplying DLL overcomes the drawbacks of PLL such as jitter accumulation, high sensitivity to supply, and substrate noise. For this reason, it represents a good performance for phase noise.

## II. Dynamic Frequency Scaling

In recent years, the power and energy consumption has become a critical design issue in the embedded systems, especially for the mobile systems and portable systems. Dynamic Voltage Frequency Scaling (DVFS) has been more important for saving energy on mobile embedded systems. Figure 24 illustrates the diagram of the voltage/frequency transition that proposed in [36]. The voltage changes from high to low and goes back to high in this example. In the conventional frequency scaling, the clock must be stopped during voltage transition. Therefore, performance overhead occurs by the frequency scaling. For the proposed frequency scaling, the voltage/frequency selectors are introduced to achieve no performance overhead as indicated in the third line of the figure.

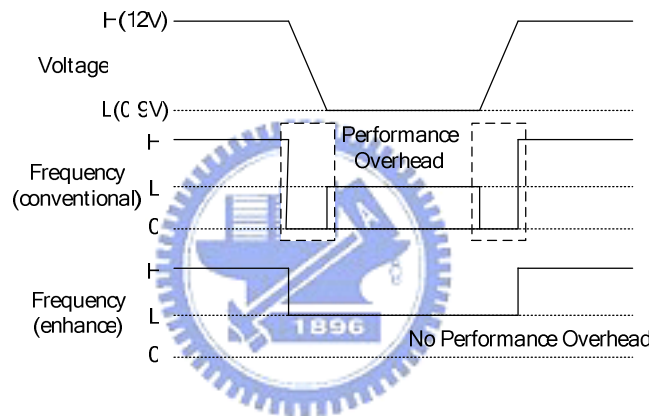


Figure 24 The operation of DVFS scheme

Notice, there are two issues need to be consideration for changing the frequency without stopping the running programs. First, the data transfer from modules operating in different frequency must be handled by the main bus. Second, the transition in supply voltage skews the clock tree.

A DVFS scheme is also proposed in [35]. A frequency adjuster circuit unit calculates the optimum clock frequency based on the activity value derived from the activity monitor to reserve the required number of inactive margin cycles within the monitoring period and indicates the next clock frequency to the clock generator. The dynamic frequency is selected by the clock thinning circuit which collects several different frequency input. Therefore, it can operate continuously without PLL relock or system.

In order not to make performance overhead, the relock time is an important issue for the DFS. All the previous mentioned DVFS schemes utilize multiple existing frequencies to generate the desired frequency. However, it increases the consumption for the useless frequency. A multiphase DLL based clock generator for dynamic frequency scaling was proposed in [31]. With plain digital logic for frequency adjustment, the multiplication factor can be changed with fast lock time. For the specific case, it only takes one-cycle to lock during frequency scaling.

### III. Transmitter [48]

In the digital communication application, the multiphase DLL can apply to a data channel compression transceiver. The architecture of the transceiver is shown in Figure 25. The transmitter's output, TX\_DATA and TX\_CLK, are sent to the receiver's inputs, RX\_DATA and RX\_CLK, respectively. In the transmitter, the generated seven-phase clock signals are used to transfer 7-bits data (DATA [6:0]) into one data channel (TX\_DATA), and the TX\_CLK is also sent to the receiver. The receiver shown in Figure 25(b) recovers the received data stream (RX\_DATA) back to original 7-bits data (DATA\_OUT [6:0]). The two-phase ADMCG shown in Figure 25(b) is used to estimate the accurate delay of  $T_{REF}/14$ . It aligns two adjacent phases of the seven-phase DLL outputs (i.e., P6 and P0) to measure the delay, and the received data stream will first be delayed by and then sampled by the seven-phase multiphase clock signals. Thus, those multiphase clock signals can sample the received data stream in the center of the bit symbol boundary, and this maximizes the timing margin of the receiver circuit

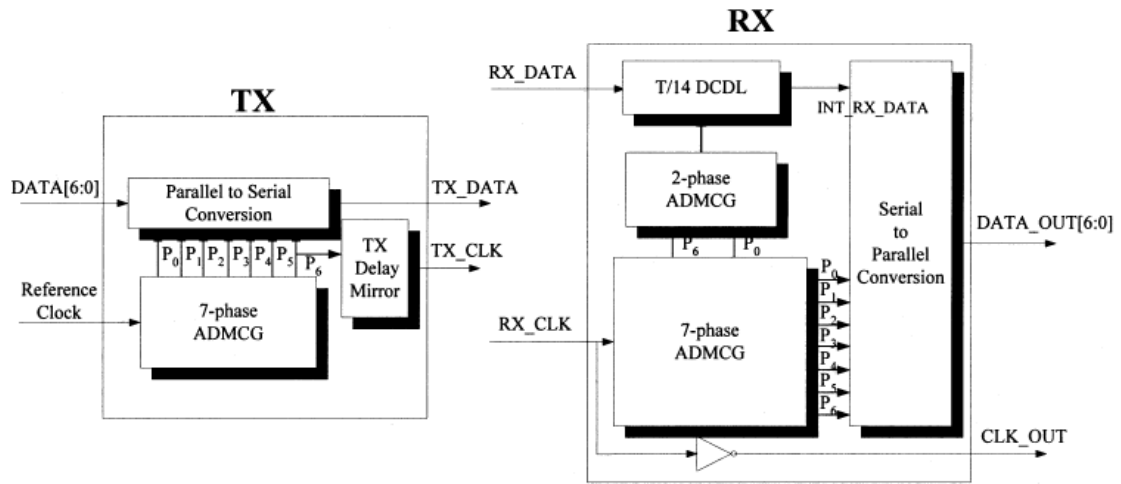


Figure 25 7:1 Data channel compression transceiver. (a) Transmitter circuit. (b) Receiver circuit.

#### IV. DDR SDRAM controller application [10]

In [10], the calculations for timing budget show that the optimal value for  $t_{SD}$  is approximately 20 percent of an input clock period as shown in Figure 26. Since the input clock frequency range from 100MHz to 200MHz (DDR-200/266/333/400), the  $t_{SD}$  value varies from 2ns ( $=10ns \times 0.2$ ) to 1ns ( $=5ns \times 0.2$ ). Therefore, a five-phase all-digital DLL was proposed in [10] to generate the desired  $t_{SD}$  delay for DQS signal.

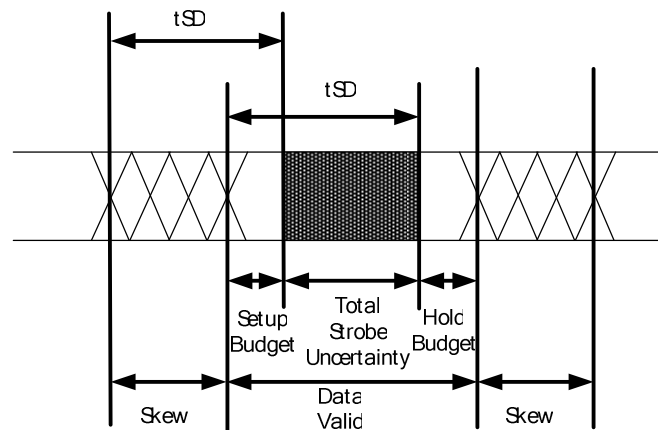


Figure 26 Read operation timing budget

The block diagram of the five-phase all-digital DLL for DDR SDRAM controller application is shown in Figure 27. Like most of DLL-based multi-phase clock generators, the DLL has a multi-stage delay line with the same control word to

generate equally spaced multi-phase clock output. It uses the time-to-digital (TDC) scheme to lock whole loop. Hence, a design consideration should be noticed is that sometimes it is difficult to meet the minimum delay constraint when using standard cell to build up a high resolution delay cell. Therefore, the DLL in this design is lock to two periods of the reference clock period by using TDC scheme. After DLL is locked, the phase spacing of each delay stage should be  $2 \cdot T_{\text{FREF}}/5$ , where  $T_{\text{FREF}}$  means the clock period of the reference clock. Hence the minimum delay constraint for each delay stage is extended twice as original. The total delay from DQS to DQSD becomes  $1.2 \cdot T_{\text{FREF}}$ , which means the phase shift between DQS and DQSD is still  $0.2 \cdot T_{\text{FREF}}$ . As a result, the desired tSD delay can be generated by the multiphase DLL

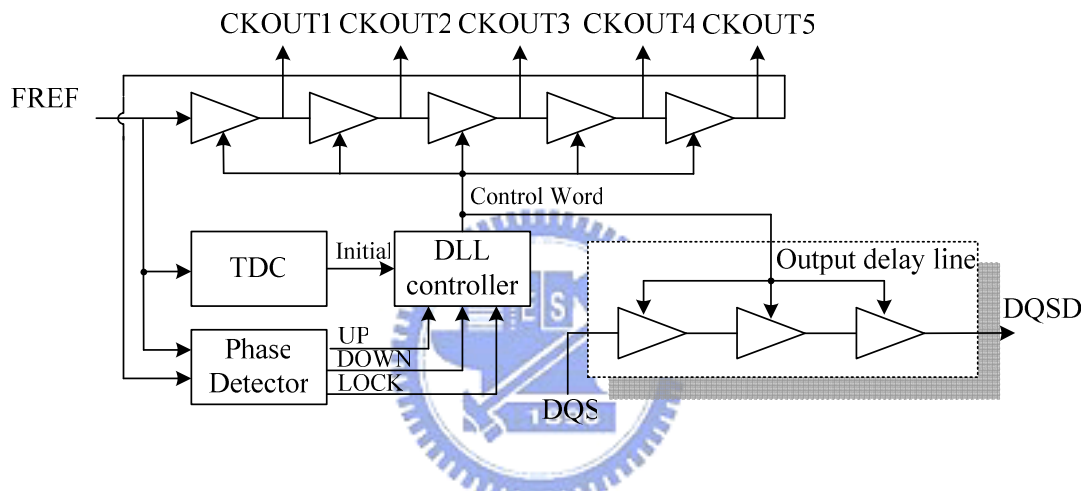


Figure 27 The block diagram of multiphase DLL for DDR SDRAM application

### 3.3 SELF-CALIBRATION TECHNIQUES

As we mention above, the multiphase clocks are useful in many applications. The feedback loop guarantees the whole loop to hold the lock state. However, each delay cell may introduce different delay time due to the process variations or wiring mismatch. It is impossible to equal each phase difference of output signals without any calibration schemes. Figure 28 shows the 1000 points Monte-Carlo simulation results for static timing errors among five delay cells, where the designed delay time is 1ns and 15 mV threshold voltage mismatch of the delay cells are added. In our 90 nm CMOS technology, the threshold voltage mismatch of 15mV for the delay cells will cause the maximum delay time mismatch of 100ps (around 10% delay mismatch). Similarly, Figure 29 shows the Monte-Carlo simulation results for the 10% channel



length mismatch. The simulation results indicate the channel length mismatch will cause the maximum delay time mismatch of 40ps (around 4% delay mismatch) for the delay cells.

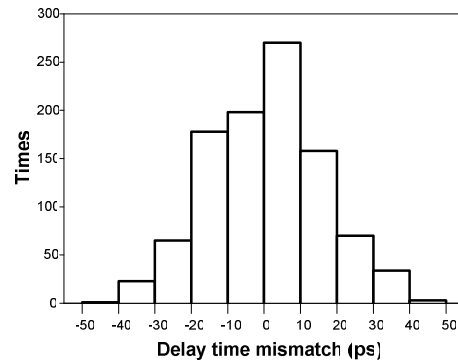


Figure 28 Delay time mismatch due to the delay cell with the threshold voltage mismatch of 15 mV

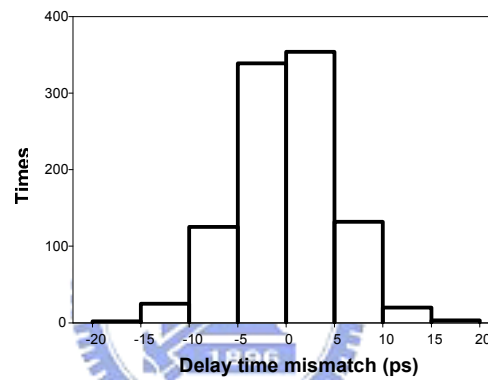


Figure 29 Delay time mismatch due to the delay cell with the channel length mismatch of 10%

Since the delay cells introduce different delay time due to the process mismatch, the additional calibration mechanisms is needed. In order to compensates the mismatches among delay cells in the DLL or PLL. One of the solutions to reduce the mismatch is to increase the transistor size. Starting from a circuit that has been optimized with respect to specifications other than noise and mismatch, one can scale the width of every component of that circuit by a certain factor  $a$  . For a delay cell, the implication of the impedance level scaling is that increasing the power by a factor  $a$  yields a stochastic jitter reduction of  $\sqrt{a}$  . Also the mismatch of the delay between different cells will improve by a factor  $\sqrt{a}$  [1].

However, impedance level scaling [1] will increase parasitic capacitance, power and area. When the clocking speed increases, the delay cell with minimum channel lengths may be chosen for the sake of higher speed [7]. Such a delay cell suffers from poor matching which may induce significant timing errors. Thus, the extra

self-calibration algorithm and its circuits for the precise multiphase DLL or PLL is necessary.

### 3.3.1 A Self-Calibration Delay-Locked Delay Line

One of self-calibration algorithms was proposed in [26], the basic concept is shown in Figure 30, where  $NDL_i$  is the differential non-linearity of  $i$ th delay cell and  $R_i$  is the contents of  $i$ -th register and  $R_{oi}$  its value at the beginning of the non-linearity test. With perform a complete code-density test with balanced mean method [26], the correction is done by comparing the register content with two thresholds that define as  $\pm 1\%$  non-linearity value and if an arithmetic overflow (or underflow) of the register is detected during the test, a interrupt is occurred for the delay cell and the cell controller ignores further hits.

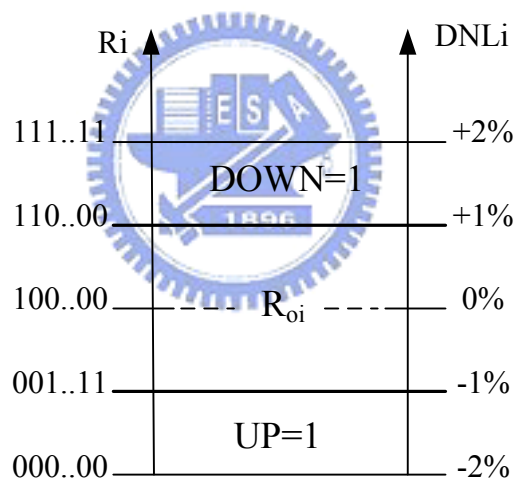


Figure 30 Self-calibration delay locked delay line scheme

The calibration produce is dependence on the most two significant bits of the register content. When these two bits are value '00' or '11' the relevant threshold has been exceeded and cell controller should be decreasing or increasing the delay time of delay cell by adjust the calibration control word. Thus the comparator can reduce to a very simple structure that consist of two logic gate and apply to a four bits up/down counter which generates the calibration control word. According to the test result, assures us that the delay mismatch of each delay cell could be pushed below 1%.

The calibration algorithm uses a time-measurement method to reach self-calibration delay locked delay line. However, there is a restriction in which if the initial non-linearity of the delay cell is out of the allowable correction range, the calibration mechanism must halt.

### 3.3.2 Sequential Phase Adjustment Calibration Technique

Another method proposed in [3] [7] avoids the mentioned problem. In [3], the operation of self-calibration, as shown in Figure 31, assume the initial phase differences between *out1* and *out2*, *out2* and *out3*, *out3* and *out4*, and *out4* and *out1* are  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , and  $\theta_4$  respectively and the target phase differences are all 90-degrees. First, the calibration produce is controlled by loop enable signal *loop\_enabli*, i.e. the *loop\_signal1* signal becomes high and selects three output signals, *out1*, *out2*, and *out3*. The selected signals are inserted into relative phase detector as shown in Figure 32 and generate a control signal *vcon1* and *von2* to adjust the phase of *out2* by changing capacitance of two adjacent capacitors. By performing relative phase comparison method the phase differences between *out1* and *out2*, *out2* and *out3* become same. Figure 31 (b) shows the result. Similarly, when next loop enable signal, *loop\_signal2*, is high, the phase differences between *out2* and *out3*, *out3* and *out4* become same as shown in Figure 31 (c). By continuously, the phase difference between *out1* and *out2*, *out2* and *out3*, *out3* and *out4*, and *out4* and *out1* become all the same, finally. Figure 31 (f) shows the final state of the DLL loops.

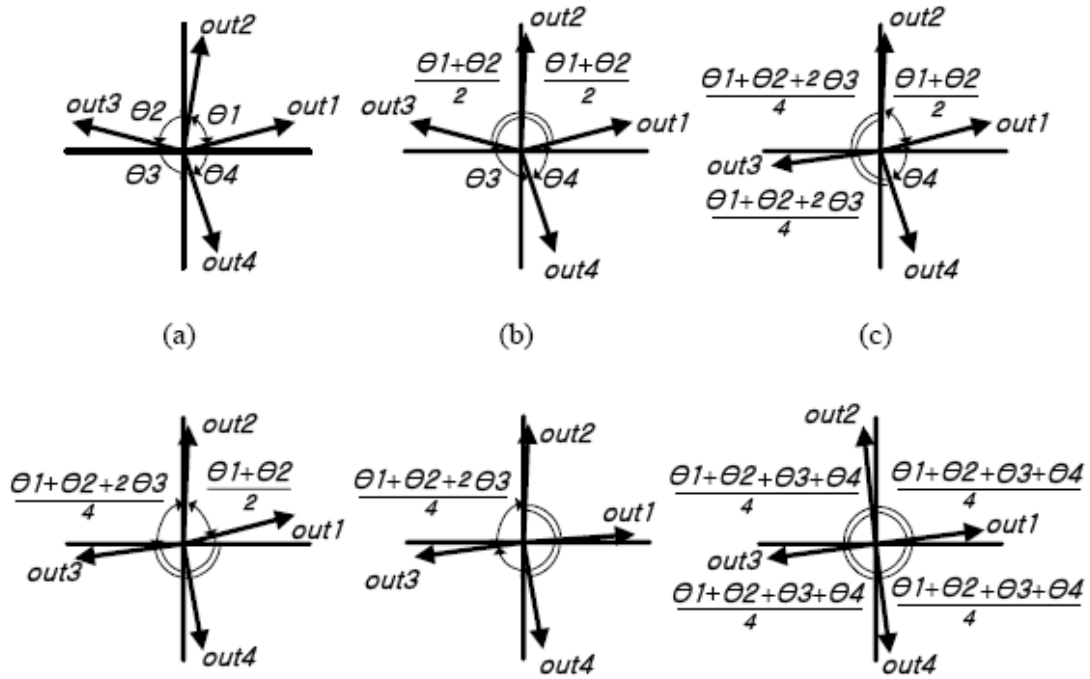


Figure 31 Sequential phase adjustment calibration algorithm

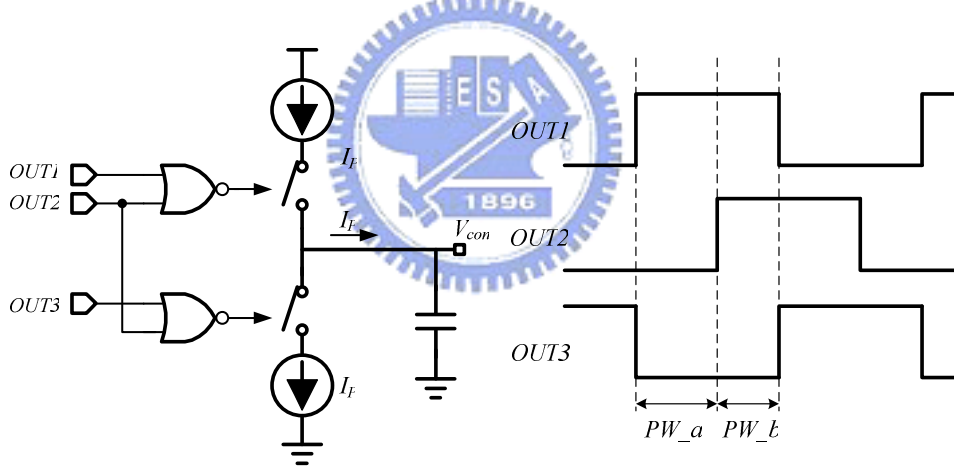


Figure 32 (a) Relative phase detector (b) Relative comparison method

### 3.3.3 Parallel Phase Adjustment Calibration Technique

In order to adjust every output phase independently and not interfere with main loop, the additional delay adjustment outside the ring oscillator or the delay line method was proposed in [2]. Figure 33 shows the parallel phase adjustment calibration algorithm. Since  $\phi_1$  is tracing reference clock by the main loop,  $\phi_5$  can be calibrated by comparing  $\Delta t_{d15}$  and  $\Delta t_{d51}$ , and  $\phi_1$  is used as the reference signal of  $\phi_5$ . When  $\phi_1$  and

$\phi_5$  is established,  $\phi_3$  can be calibrated by comparing  $\Delta td_{13}$  and  $\Delta td_{35}$ , and  $\phi_7$  by  $\Delta td_{57}$  and  $\Delta td_{71}$ . This process is repeated for the other phases and finally the phase differences of each delay cell can be calibrated.

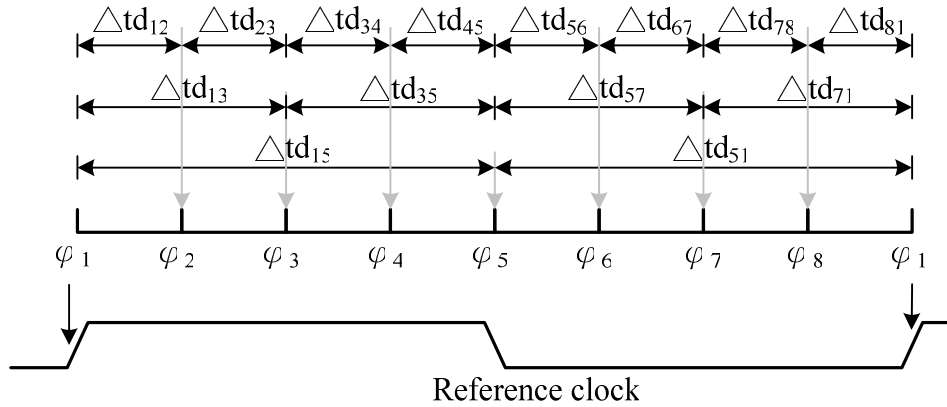


Figure 33 Parallel phase adjustment calibration algorithm

Figure 34 shows the circuit design to implement the parallel phase adjustment calibration algorithm that we mention above. The delay sensing circuit, as shown in Figure 34 (a), is used to produce the time delay pulse width,  $\Delta td_{ij}$ . The calibration loop charge pump, as shown in Figure 34 (b), is a simple current-steering structure and the capacitor is implemented with pMOS transistor.

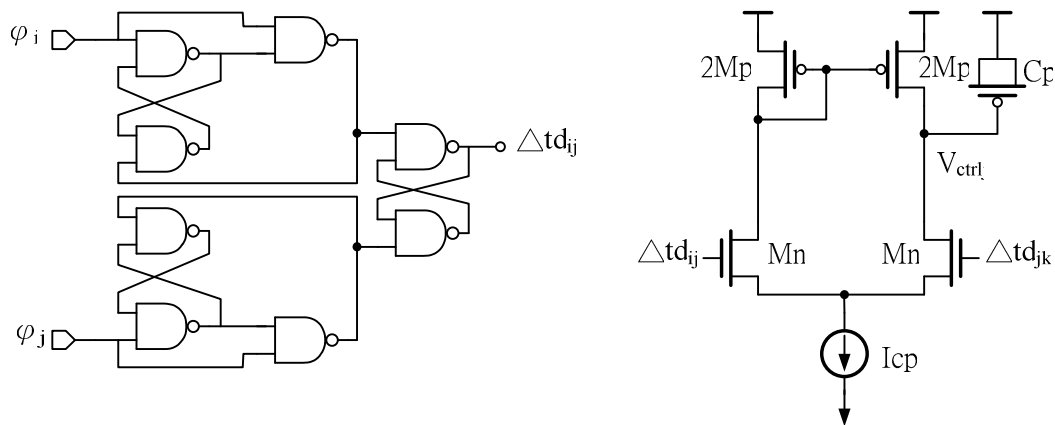


Figure 34 (a) Delay sensing circuit. (b) Calibration loop charge pump

### 3.3.4 A PLL BASED SELF-CALIBRATED ALGORITHM

The self-calibrated algorithm which proposed in [25] is based on the innate ability of PLL. Assume the fractional-N frequency synthesizer is based on a PLL capable of generating 8 different phase clock signals. Each edge of the clock signal is used to synthesize the output signals. Consequently, the division ratio becomes  $(M+1/8)$ . When the PLL is locked, the amount phase error  $\Delta t_i$  that caused by the delay mismatches becomes zero, where  $i$  means the  $i$ th delay cell. In other word,

$$Vt_1 + Vt_2 + \dots + Vt_8 = 0 \quad (3.1)$$

Assume the phase offset of the 1st delay cell is change by  $\Delta^1_1$ . The after the PLL is locked again, the resulted phase offset becomes:

$$Vt_1^1 = Vt_1 - \frac{Vt_1^1}{8}, Vt_2^1 = Vt_2 + \frac{Vt_1^1}{8}, Vt_8^1 = Vt_8 + \frac{Vt_1^1}{8} \quad (3.2)$$

Where  $\Delta^k_N$  is the phase error due to  $N$ -th delay cell after  $K$  cycles of calibration, and  $\Delta^m_N$  is the amount of the calibration at the  $m$ -th iteration.

By repeating the above step for each delay cell one by one until

$$\Delta^m_N = 0 \quad (3.3)$$

is satisfied for all delay cells, the final values of the phase error due to 1st delay cell becomes;

$$\begin{aligned} Vt_1^{final} &= Vt_1 - \sum_{k=1}^7 \frac{1}{8} Vt_1^k + \sum_{k=1}^7 \frac{1}{8} Vt_2^k + \dots + \sum_{k=1}^7 \frac{1}{8} Vt_8^k \\ &= Vt_1 - Vt_1 + \frac{1}{8} \sum_{k=1}^7 Vt_n = 0 \end{aligned} \quad (3.4)$$

Similarly,

$$Vt_1^{final} = Vt_2^{final} = Vt_3^{final} = \dots = Vt_8^{final} = 0 \quad (3.5)$$

Therefore, all the phase errors due to the delay mismatches are reduces to zero when the compensation algorithm is finished.

### 3.4 RAPID SELF-CALIBRATION ALGORITHM

In [26] [3] [7] [2] [25], the calibration algorithm has been adopted to overcome PVT variations. The self-calibration algorithm [3], [7] requires additional timing control circuit, and large calibration cycles. A novel rapid self-calibration (RSC) algorithm was proposed to reduce calibrate cycle, where no extra timing control circuits is needed.

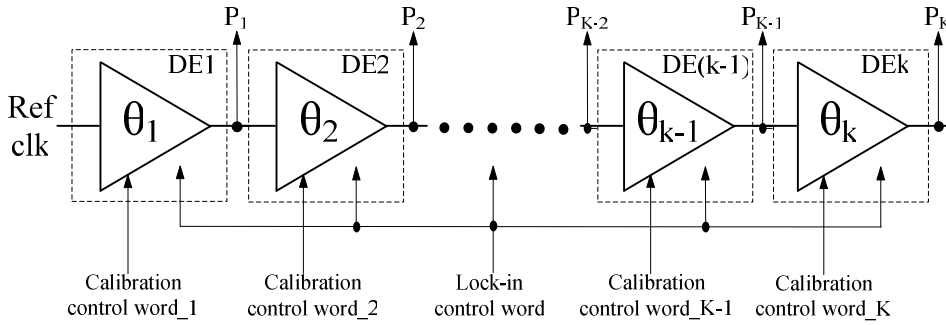


Figure 35 the idea of proposed RSC algorithm

Figure 35 shows the idea of proposed RSC algorithm. Assume the multiphase DLL is consisted of  $k$  identical digitally controlled delay elements and multiphase output generated from each delay element. Unlike conventional multiphase DLL which the delay elements are controlled by a global signal, each delay element in the proposed RSC algorithm is controlled by two sets of control words. They are lock-in control word and calibration control word $_i$ , where  $i$  means the  $i$ th delay stage. The lock-in control word is connected to all delay elements and each delay element has distinct calibration control word $_i$  as shown in Figure 35.

In the beginning, the total delay of the DCDL locks to multiple period of reference clock,  $Ref_{clk}$ , by changing the lock-in control word. After the DLL is locked, i.e. the phase difference between  $Ref_{clk}$  and last output signal,  $P_k$ , is equal to one reference clock period. The RSC algorithm first considers about three signals; they are  $Ref_{clk}$ ,  $P_1$ , and  $P_2$ . A relative comparison method [3] is adopted to adjust  $\theta_1$  to  $(\theta_1+\theta_2)/2$  by changing the calibration control word $_1$ , where  $\theta_i$  means the phase difference between  $P_i$  and  $P_{i-1}$ . Similarly, the calibration unit would consider about the next three signals; they are  $P_1$ ,  $P_2$ , and  $P_3$ . It will adjust  $\theta_2$  to  $(\theta_2+\theta_3)/2$  by changing the calibration control word $_2$ . Unlike [3], the modified  $\theta_1$  does not affect  $\theta_2$ . This

allows sequential adjustment in the same reference cycle. Finally, the adjustment of  $\theta_k$  is based on  $P_k$  and  $Ref\_clk$ , which guarantees the whole DLL remains locked. The lock-in control word remains unchanged during the calibration process to ensure successful RSC operation. As the result, the final output difference of each delay stage is one fifth of the period of reference clock.

Figure 36 shows the RSC algorithm expressed in mathematical equations when the delay stages number is five. Assume the DLL is in the locked state initially and fulfill equation (3.6).

$$q_1 + q_2 + q_3 + q_4 + q_5 = 360^\circ \quad (3.6)$$

In the first calibration cycle, i.e.  $n=1$ , the  $\theta_1$  becomes the mathematical average value of  $\theta_1$  and  $\theta_2$ , which is expressed as

$$\frac{q_1 + q_2}{2} \quad (3.7)$$

Where  $\theta_j^i$  means the phase error due to  $j$ -th delay cell after  $i$ -th calibration cycles, and  $\theta_1$  and  $\theta_2$  are the initial phase differences between  $Ref\_clk$  and  $P1$ ,  $P2$  and  $P3$ . At the same calibration cycle, the phase difference between  $P2$  and  $P3$ ,  $\theta_2$ , becomes

$$\frac{q_2 + q_3}{2} \quad (3.8)$$

The phase adjustment of  $\theta_3$  and  $\theta_4$  in the first calibration cycle is similar to  $\theta_1$  and  $\theta_2$ . Because the DLL is hold in the locked state,  $\theta_5$  would not change at the first calibration cycle. Next, when  $n=2$ , the phase differences  $\theta_1$  and  $\theta_2$  become equation (4) and equation (5), respectively.

$$\frac{q_1 + 2q_2 + q_3}{4} \quad (3.9)$$

$$\frac{q_2 + 2q_3 + q_4}{4} \quad (3.10)$$

Note that after the first calibration cycle, the equation (1) may not be observed. In order to guarantee the whole DLL remains locked,  $\theta_5$  becomes



$$\frac{q_5 + q_1}{2} \quad (3.11)$$

As the n increases as any number N, one of the phase differences can be express as

$$\frac{Aq_1 + Bq_2 + Cq_3 + Dq_4 + Eq_5}{2^N} \quad (3.12)$$

Where the sum of A, B, C, D and E is  $2^N$ . If the N goes to infinity, each phase differences can be expressed as

$$\frac{(2^{\infty} / 5)(q_1 + q_2 + q_3 + q_4 + q_5)}{2^{\infty}} = \frac{q_1 + q_2 + q_3 + q_4 + q_5}{5} \quad (3.13)$$

Equation (3.13) shows the phase differences of each delay stage become identical and fulfill equation (3.6) means that the DLL remains locked. As a result, the calibration algorithm guarantees the final phase difference of multiphase DLL are all  $360^\circ/5$ . Figure 36 shows the step summary of proposed RSC algorithm.

n=0	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$
n=1	$\frac{\theta_1 + \theta_2}{2}$	$\frac{\theta_2 + \theta_3}{2}$	$\frac{\theta_3 + \theta_4}{2}$	$\frac{\theta_4 + \theta_5}{2}$	$\theta_5$
n=2	$\frac{\theta_1 + 2\theta_2 + \theta_3}{4}$	$\frac{\theta_2 + 2\theta_3 + \theta_4}{4}$	$\frac{\theta_3 + 2\theta_4 + \theta_5}{4}$	$\frac{\theta_4 + 3\theta_5}{4}$	$\frac{\theta_5 + \theta_1}{2}$
n=3	$\frac{\theta_1 + 3\theta_2 + 3\theta_3 + \theta_4}{8}$	$\frac{\theta_2 + 3\theta_3 + 3\theta_4 + \theta_5}{8}$	$\frac{\theta_3 + 3\theta_4 + 4\theta_5}{8}$	$\frac{\theta_4 + 5\theta_5 + 2\theta_1}{8}$	$\frac{3\theta_1 + \theta_2}{4}$
n=4	$\frac{\theta_1 + 4\theta_2 + 6\theta_3 + 4\theta_4 + \theta_5}{16}$	$\frac{\theta_2 + 4\theta_3 + 6\theta_4 + 5\theta_5}{16}$	$\frac{\theta_3 + 4\theta_4 + 9\theta_5 + 2\theta_1}{16}$	$\frac{\theta_4 + 5\theta_5 + 8\theta_1 + 2\theta_2}{16}$	$\frac{5\theta_1 + 4\theta_2 + \theta_3 - 2\theta_5}{8}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n=N	$\frac{A\theta_1 + B\theta_2 + C\theta_3 + D\theta_4 + E\theta_5}{2^N}$	$\frac{A'\theta_1 + B'\theta_2 + C'\theta_3 + D'\theta_4 + E'\theta_5}{2^N}$	$\frac{A''\theta_1 + B''\theta_2 + C''\theta_3 + D''\theta_4 + E''\theta_5}{2^N}$	$\frac{A'''\theta_1 + B'''\theta_2 + C'''\theta_3 + D'''\theta_4 + E'''\theta_5}{2^N}$	$\frac{A''''\theta_1 + B''''\theta_2 + C''''\theta_3 + D''''\theta_4 + E''''\theta_5}{2^{N-1}}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n=∞	$\frac{\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5}{5}$	$\frac{\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5}{5}$	$\frac{\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5}{5}$	$\frac{\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5}{5}$	$\frac{\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5}{5}$

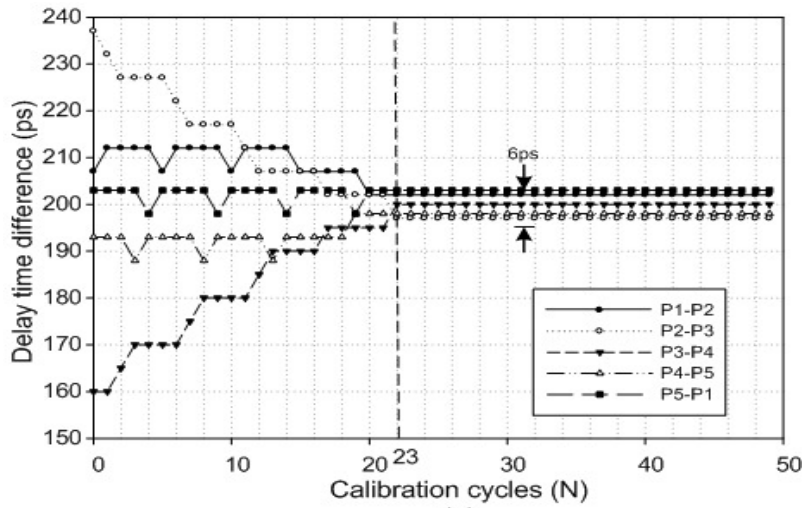
Figure 36 Step summary of RSC algorithm

Since the calibration circuit performs in the digital domain, the target is approached based on unit step  $\Delta T$ . However, the quantization error,  $q_e$ , is unavoidable. In [7], the calibration mechanism modified the phase difference by adjusting the control word of the output stage buffers, where the last stage of the output buffers is fixed to ensure whole loop stays locked. The mechanism, however, increases the maximum phase error.

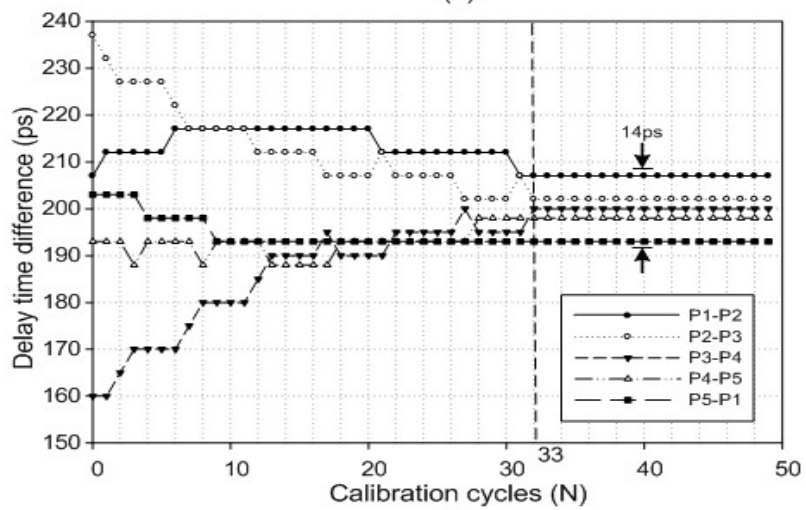
Figure 37 shows the calibration step simulation results for the IQ-style [3], buffer style [7], and proposed RSC algorithm. The simulation environment setting is: the

number of delay stages is five, unit step  $\Delta T=5ps$ , quantization error= 5ps, the period of reference clock is 1ns and 20% random delay mismatch is adopted. Base on these algorithms, the calibration produces is converged. The calibration cycles are 23, 33, 20 and maximum phase difference between any two output signals are 6ps, 14ps, 6ps for IQ-style, buffer style, and proposed RSC algorithm, respectively. As the result, the proposed RSC algorithm achieves the fastest calibration time and smaller phase error between any two outputs.

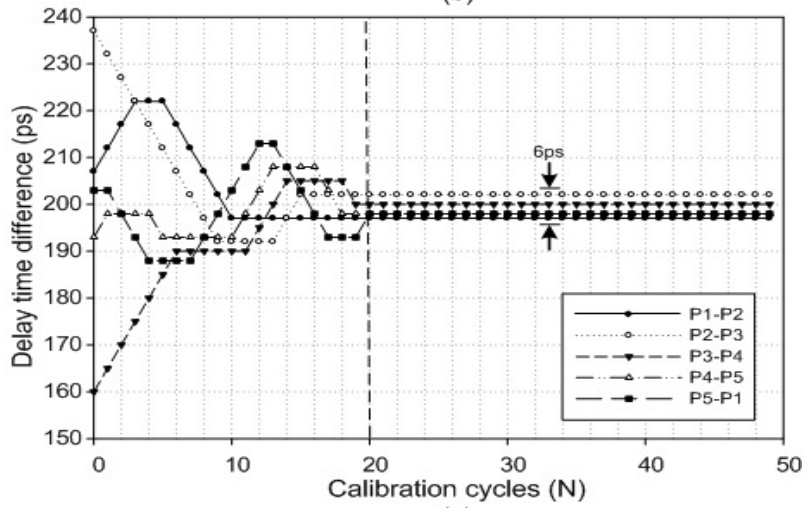




(a)



(b)



(c)

Figure 37 The calibration produce comparison with (a) IQ-style (b) buffer-style (c) RSC-based

Figure 38 shows another simulation result for the calibration cycles comparison between the buffer style [7] and RSC based calibration schemes. The environment

setting is the delay stages number is five, unit step and quantization error are both 5ps.

The proposed RSC algorithm has up to 4.17x speed improvement.

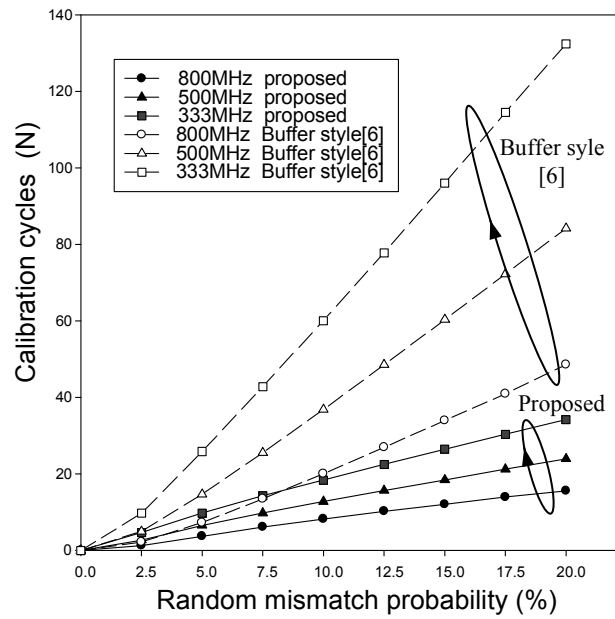


Figure 38 Calibration cycles comparison



# CHAPTER 4

## A WIDE-RANGE AND FAST-LOCK ALL-DIGITAL DELAY-LOCKED LOOP

In this chapter, an unbalance binary search algorithm is proposed to achieve short lock time and extend the locking range. An unbalance binary search algorithm based (UBS) controller is designed and it is suitable for all-digital DLL design. Compare with the conventional counter-controlled DLL (CDLL), successive approximation register-controlled DLL (SARDLL), and variable SAR DLL (VSARDLL), a UBS-controller based DLL can achieve fast lock time. Moreover, the harmonic-locking issue can be avoided in the wide-range operation. The circuit implementation is based on the UMC 90 CMOS technology model, and the operating range is 100MHz to 500MHz with 1V supply voltage.

### 4.1 INTRODUCTION OF WIDE-RANGE DLL

Phase-Locked loops (PLL) and delay-locked loops (DLLs) have been widely adopted to eliminate clock signal skews and jitter in high-speed microprocessors, memory interfaces and communication integrated circuits (ICs). Generally, the DLL has better jitter performance than the PLL because there is no jitter accumulation characteristic in the DLL. In the past, the locking time is an important parameter of the digital DLL performance. Conventional digital DLL, such as RDLL [14] and CDLL, the locking time and the number of the delay cells increase exponentially as the number of control bits increases. The successive approximation register-controlled [4] DLL reduces the locking time by using the binary search algorithm. The time-measurement (TM) DLL use the time-to-digital converter (TDC) to achieve the shortest locking time, however, the area requirement and power consumption is the cost. Compare with these

DLLs, the SAR controlled may be the most suitable for DLL by balancing the power consumption, area requirement and locking time.

Recently, the low-power issue replace the high-speed issue becomes the most concernment in the system on chip (SoC) applications. The DLL with wide operation range is especially required in the SoC systems which support dynamic operation frequencies and variable supply voltage scaling techniques to achieve better performance with low power consumption. Unfortunately, the DLL may suffer harmonic locking over a wide operation range.

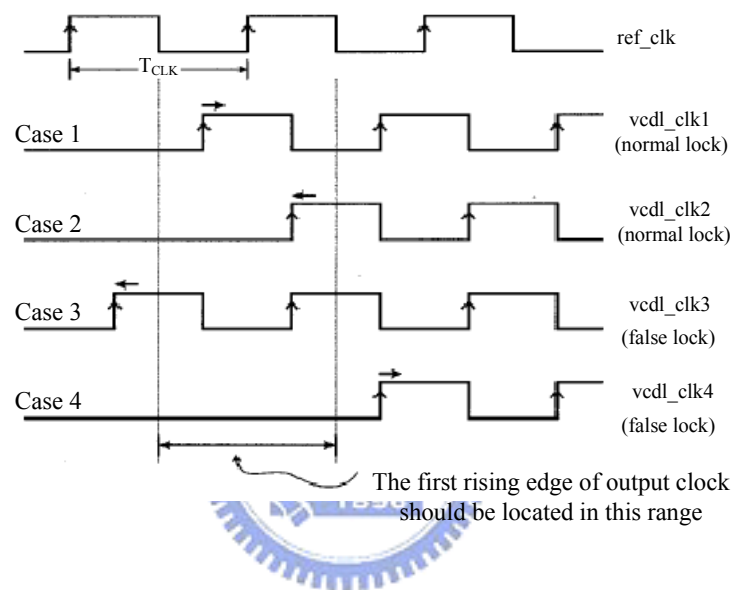


Figure 39 Harmonic locking problems.

Figure 39 shows the harmonic locking problem [12]. In the normal condition (case 1 and case 2), the DLL locks the delay which is just one clock cycle of input reference signal. However, the conventional DLL will fail to lock (case 3) or falsely lock to two or more periods (case 4),  $T_{clk}$ , of the input signal if the initial delay of the VCDL is shorter than  $0.5 \cdot T_{clk}$  or longer than  $1.5 \cdot T_{clk}$ . Consequently, the additional control block or locking mechanism is needed to avoid the false lock. Next Section will describe the previous research of wide range scheme.

## 4.2 PREVIOUS RESEARCH OF WIDE-RANGE DLL

Various wide-range DLLs have been proposed [30], [52], and [19] to solve the harmonic lock problem. The DLL with a false-lock capability phase detector [52] is

used to prevent the harmonic lock. Figure 40 (a) shows the phase detector. The multiphase outputs of the DLL are applied to the phase detector. When the initial delay of VCDL is larger than  $2 \cdot T_{clk}$ , i.e., the rising edge of  $D_1$ , a clock delayed by  $1/8 T_{VCDL}$ , a *coarse\_up* signal is asserted to reduce the delay time of VCDL to prevent harmonic lock. On the other hand, if the initial delay of VCDL is so small that causes the delay time stuck its minimum value, the *PD\_reset* signal is asserted to deactivate the false up signal. Figure 40(b) shows the timing diagram of the false-lock capability phase detector. In [6], an all-analog DLL uses the replica delay line and a cycle period detector to solve the false lock and narrow operating frequency-range problem of a conventional DLL. The auxiliary loop not only uses as a replica delay line but also monitors the lock state of main loop by estimating the cycle period of the input clock. However, the he analog DLLs are more sensitive to the process variation. Therefore, digital DLL is developed to improve process portability.

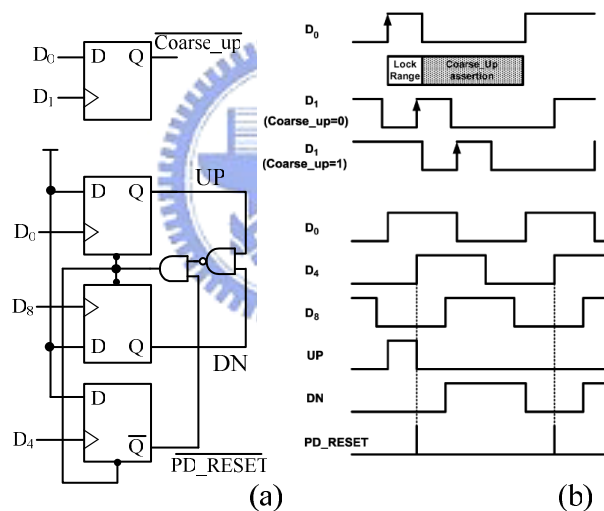


Figure 40 (a) False-lock capability phase detector and its (b) timing diagram.

In [19], phase selector circuit and start-controlled circuit are used to solve false locking problems enlarge the operating frequency range. The phase selection circuit will automatically select one of the delayed outputs to feedback. The start-controlled circuit controls the locking produce. First, the delay between input and output of the delay line is set to the minimum value and then increases the delay of delay line until it reaches one clock period of the input clock. The timing of the external start-up signal must be carefully designed because the external input clock of the DLL must be settled down before the initialization of the DLL starts up. If the initialization or phase

selection step is started before the input clock is stabilized, the DLL may not achieve the correct lock due to its wrong phase selection [19].

The time-to-digital (TDC) scheme may be the simplest concept to resolve the harmonic problem. In [48], the all-digital multiphase clock generator is used to overcome the false locking problem. The TDC circuit measures the period of the input clock directly and converts the timing information to digital signals to control the delay of the delay line. However, such DLLs result in complex architectures that face such problems as increased area, increased power consumption and degraded jitter performance.

The variable successive approximation register (VSAR) algorithm was proposed in [5] for the all-digital DLL applications. The control unit is composed of conventional SAR units, variable SAR units and fail-to-lock judgment circuit (FJC). Initially, the conventional SAR units borrow one bit as a MSB from the LSB of variable SAR units to perform a binary search. After the binary search is finished, the FJC examines the lock state. When the locking produce is fail, the conventional SAR units borrow one more LSB from the variable units and repeat the locking produce. Before the total number of borrowed bits is reached, the locking produce repeats until the DLL is locked correctly. Once the lock state is confirmed, the control unit is transformed into a counter for a closed-loop operation. Base on the variable algorithm, the delay of delay line increases gradually form the minimum and never exceed twice of the input clock period. Therefore, the harmonic locking problem can be avoided.

In comparison with the conventional binary search algorithm, the variable SAR algorithm has two advantages: it vary the open-loop characteristic of conventional binary search algorithm to the close-loop type. The division ratio (DR) base on the variable SAR algorithm can be the minimum of two. It accelerates the locking time; compare with conventional SAR, up to 7X speedup can be achieved. However, the most drawback of the variable SAR algorithm is the complex hardware required and more power consumption.

### **4.3 UNBALANCE BINARY SEARCH ALGORITHM**

The design considerations of the lock-in controller include lock time, lock range



and area requirement. In order to balance these considerations, a binary search based lock-in controller may be the most suitable for the all-digital DLL. However, the conventional binary search based controller has narrow operating frequency. Figure 41 shows the false lock, when the operating frequency over the lock range.

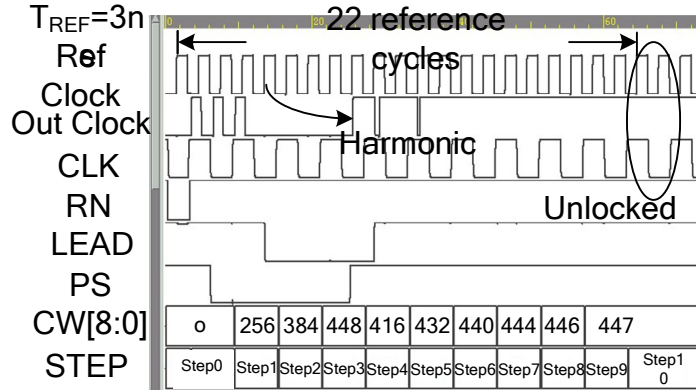


Figure 41 Conventional binary search based controller

Several different binary search algorithms were discussed in [5] and Section 4.2. In order to maximize the effective sampling rate, the DLL is desired to lock the delay which is equal to one clock period. To avoid false lock (or harmonic locking [section 4.1]), the DCDL should always operate under the delay range [12], as

$$0.5 T_{REF} < T_{DCDL} < 1.5 T_{REF} \quad (4.1)$$

where  $T_{REF}$  means the reference clock period, and  $T_{DCDL}$  means the delay time of the delay line. With  $T_{INITIAL}$  representing the initial delay time of the DLL, the relationship in equation (4.2) should be satisfied. However, the locking range of conventional binary search mechanism is restricted by equation (4.2).

$$\begin{aligned} & \text{Max}(T_{DCDL\_MIN}, \frac{2}{3} T_{INITIAL}) \\ & < T_{REF} < \text{MIN}(T_{DCDL\_MAX}, 2 T_{INITIAL}) \end{aligned} \quad (4.2)$$

The proposed unbalance binary search algorithm in this circuit is to choose an appropriate  $T_{INITIAL}$  instead of always choosing the middle point of the DCDL. Figure 42 describes the flow chart of the 3-bit unbalance binary search algorithm. In the beginning (step 0), the delay time of the delay line is adjust to the minimum and then a judge circuit exams whether the reference clock period leads twice of the minimum

delay of the delay line. If so, the next operation will go to the step1. If not, the next operation will bypass the step 1 and go to the step 2. After the step 0, no matter how the operation is, it will perform the conventional binary search until all the control bits are determined.

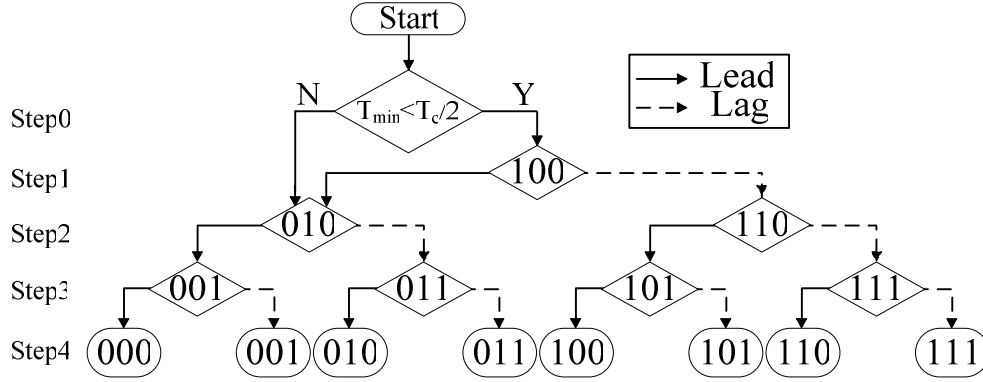


Figure 42 3-bit unbalance binary search algorithm.

For example, Figure 43 shows the difference between the conventional binary search (BS) algorithm and the proposed unbalance binary search (UBS) algorithm, where  $T_{target}$  means the targeted delay time and  $T_{initial\_i}$  means the initial delay time of the delay line. For the conventional binary search, the initial delay time is always set to the  $T_{initial\_1}$  which is equal to the average of  $T_{DCDL\_MIN}$  and  $T_{DCDL\_MAX}$ . Substitute the  $T_{initial\_1}$  into equation (4.2), the locking range is limit to (4.3).

$$(T_{DCDL\_MAX} + T_{DCDL\_MIN})/3 < T_{Target} < T_{DCDL\_MAX} \quad (4.3)$$

When the targeted delay time is out of the range of equation (4.3), the conventional binary search will fail without any hardware for the harmonic detection. However, the proposed UBS algorithm has two distinct initial delay time,  $T_{initial\_1}$  and  $T_{initial\_2}$ . When the  $T_{target}$  does not fulfill equation (4.3), it will choose the  $T_{initial\_2}$  which is equal to the quarter of  $T_{DCDL\_MIN}$  and  $T_{DCDL\_MAX}$  to be the initial delay time of delay line. Therefore, the false lock problem can be avoided. In addition, to avoid false locking in the PVT variation, the limited ranges should be overlapped and equation 4.4 should be satisfied

$$T_{DCDL\_MIN} > (2/3) T_{INITIAL\_2} \quad (4.4)$$

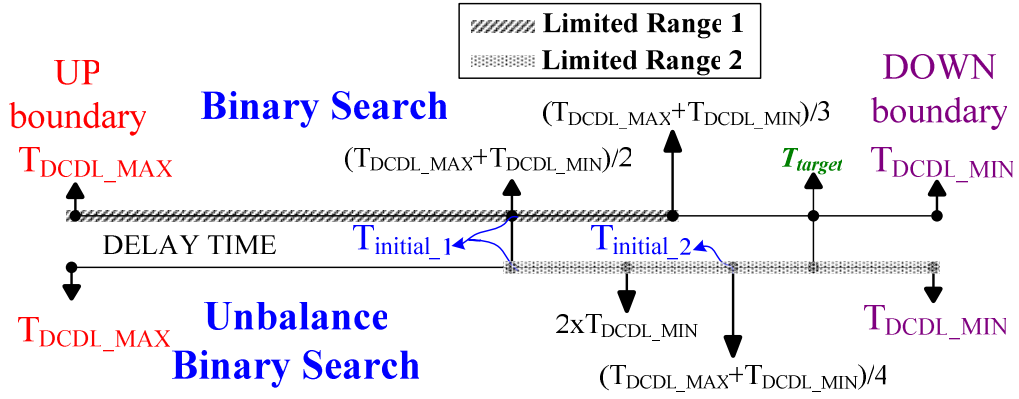


Figure 43 Comparison with conventional BS and UBS algorithm

Comparison with the conventional binary search, the proposed unbalance binary search algorithm has two advantages: one is avoiding the harmonic locking problem; the other is the improvement of the division ratio. When a wide operating range over a binary search based DLL, the minimum division ratio of two is not allowed [5]. However, it increases the locking time. On the other hand, in the proposed UBS algorithm, the division ratio could be set to two to achieve fastest lock-in time because  $T_{DCDL}$  is always fit in with equation (4.1). Assume the total control word of DCDL is  $n$ -bit and the division ratio is the minimum of two. The lock time for the DLL based on the UBS algorithm is given as.

$$T_{lock,UBS} = \begin{cases} 2^{\lceil n+2 \rceil} & \text{if } T_{REF} > 2 T_{DCDL\_MIN} \\ 2^{\lceil n+2-s \rceil} & \text{if } T_{REF} \leq 2 T_{DCDL\_MIN} \end{cases} \quad (4.5)$$

Where  $T_{DCDL\_MIN}$  means the minimum delay time of the DCDL, and integer,  $S$  means the skip step number. The extra  $2*2$  cycles are the require time to evaluate the initial state. If the reference clock period is smaller than twice of minimum delay of the delay line, it can save two reference clock cycles to lock. For example, assume that the control word needs be determined with 9-bit resolution, and the maximum and minimum delay time of the delay line is 10ns and 2ns, respectively. The  $T_{INITIAL\_1}$  is set to the middle of the provided delay range, assume 6ns in this case. The limited-range-1 can achieve 4ns to 12ns. The  $T_{INITIAL\_2}$  is set to 3ns and limited-range-2 can achieve to 2ns to 6ns. For the  $T_{INITIAL\_2}$ , the step controller can skip two steps to achieve fast-lock. The lock time is  $2 \times (9+2-2) = 18$  cycles and  $2 \times (9+2) = 22$  cycles when the reference clock period,  $T_{REF}$ , is below and over 4 ns, respectively.

Figure 44 illustrates the simulated lock time for the counter based, SAR based, VSAR based and the UBS based schemes. The proposed UBS algorithm achieves the shortest lock cycles for the most cases.

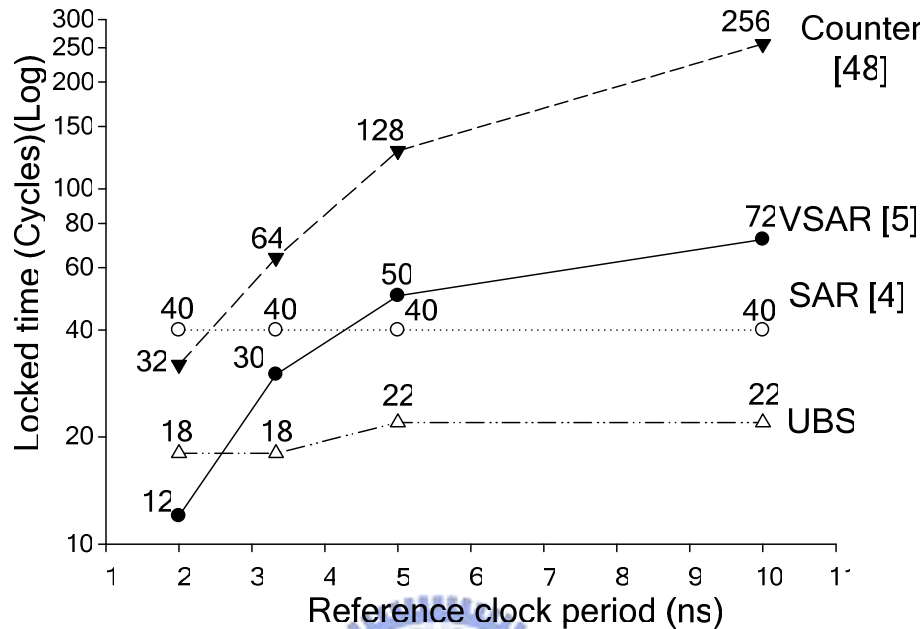


Figure 44 Simulation lock time versus the operation range.

## 4.4 CIRCUIT DESCRIPTION

The implementation of UBS algorithm based controller can simple divide into two parts. They are step controller and binary controller. The step controller would indicate which step is now entering the UBS controller. Meanwhile, the binary controller handles conventional binary search operation, where each control bit is given by the single-bit generator. Next section will describe in detail.

### 4.4.1 Step Controller

Figure 45 shows the block diagram of the step controller for the 9-bit UBS controller. It consists of a set of shift-register, a positive-edge triggered DFF, divide-by-two circuit and multiplexers. In the beginning, all the shift-registers are forced to be in the initialization state (Step 0) and the output signal (OUT) is reset to the shortest delay time. D-Flip-Flop<sub>2</sub> would tell whether  $T_{DCDL\_MIN}$  is larger than

twice of  $T_{REF}$  to decide the desired stage and which step should be entered to decide  $T_{INITIAL}$ . After decide  $T_{INITIAL}$ , the shift-register will active by the triggered signal which is the output of the divide- by-two circuit. The shift-register would store which step is and the lock signal, *LOCKED*, would be held in the end.

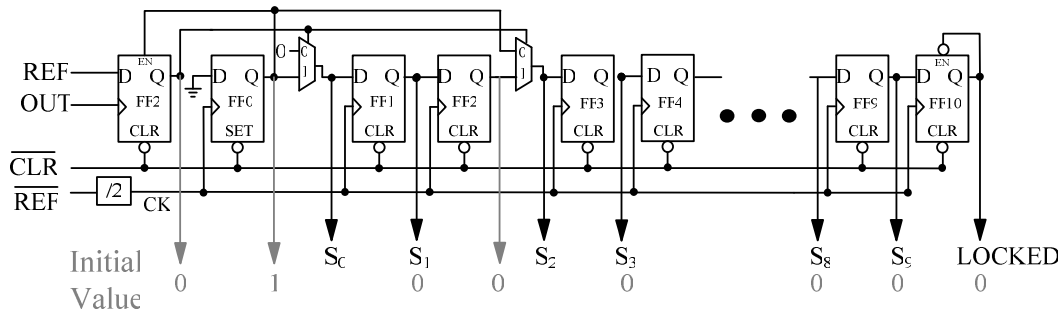


Figure 45 9-bit step controller

#### 4.4.2 Binary Controller

The architecture of the 9-bit binary controller is composed of a set of single bit generators, as shown in figure 46. Figure 46(b) shows the signal bit generator (SBG) which is multiple input register and the corresponding truth table. It consists of a positive-edge triggered DFF, multiplexers, and some logic circuits. The SBG have three different data inputs coming from: 1. the outputs of step controller,  $S_i$  and  $S_{i-1}$  where  $i$  means the  $i$ -th SBG; 2. the output phase detector, LEAD; and 3. the output of the  $i$ th SBG itself. The operation of SBG is based on the truth table in figure 46(b) to perform a binary search.

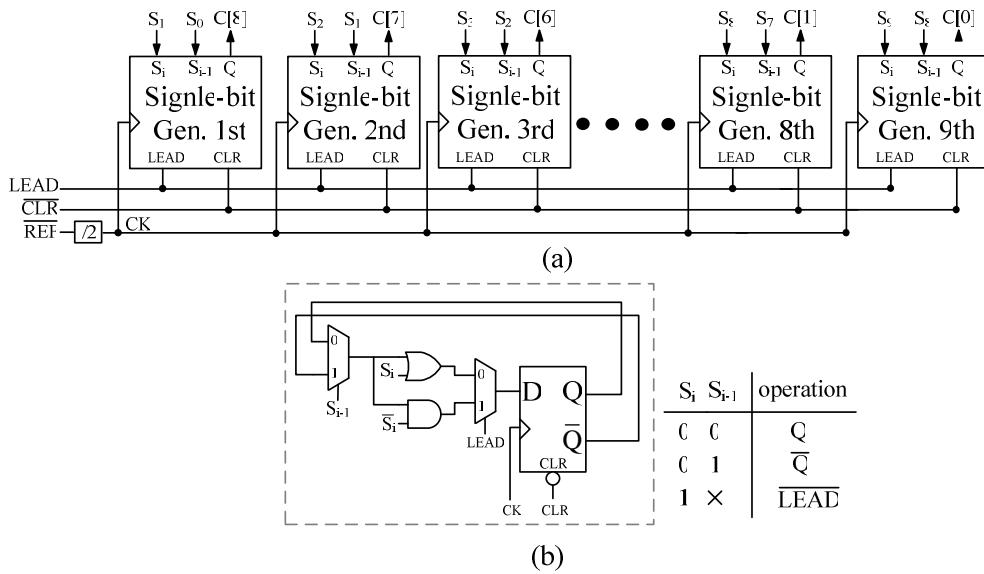


Figure 46 (a) 9-bit binary controller (b) SBG

For example, figure 47 shows the simulation result of the UBS controller when the reference clock period is 3ns. In the beginning (Step 0), the delay time is set to minimum value which equals to 2ns in this case. The D-Flip-Flop\_2 would tell whether the period of reference clock is larger than twice of the minimum delay and produces the comparison result,  $PS$ . In this case,  $PC$  is 0. Base on the result of  $PC$ , the initial delay time,  $T_{INITIAL}$ , would be set to quarter of provided delay time (3ns) and entering to the step 3. After entering Step 3, the control word becomes {001000000}, since  $C[6]$  is “guessed” as 1. The phase detector examines whether the output clock leads the reference clock and generates the output signal,  $LEAD$ . Assume that  $LEAD$  is 0, i.e., the output clock lags the reference clock. Thus, the “guessed” 1 should be cleared otherwise the delay will be too long, and keep on guessing for the next bit. It will change the control word to {000100000} and entering the sequential step when the triggered signal,  $CLK$ , is coming. If the  $LEAD$  is 1, means that the exist delay time which provided by the delay line is not enough. Therefore, the guessed bit will be held and repeat the produces until finish the LOCK signal is asserted. The total lock time is 18 reference clock cycles.

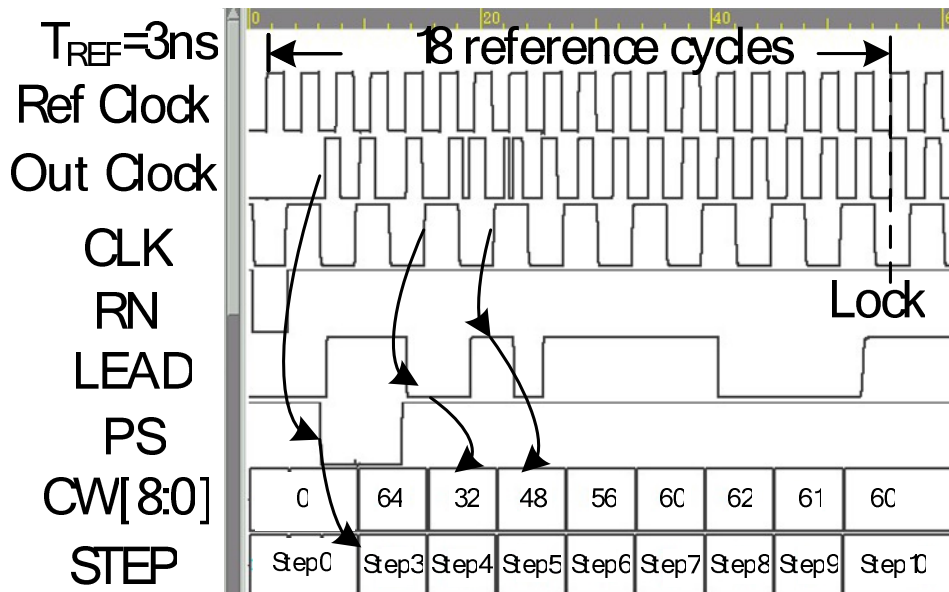


Figure 47 The operation of the UBS controller.

### 4.4.3 Digitally Controlled Delay Line

The power consumption, linearity characteristic, delay resolution, and tolerance to PVT variations are the main design considerations of the digitally controlled delay line (DCDL). In [23] [21] [13], a binary-weighted differential-delay cell (BWDC) was proposed to achieve the main consideration. There are two distinct features of BWDC that contribute to low power. First, there is no need for large driving and so logic gates can be minimally sized. Second, the de-multiplexing gates are placed at the input side so that only the components in one path are activated [13].

The architecture of the DCDL can be divided into two parts: coarse stage and fine stage as shown in Figure 48. The DCDL control word (CW) is the 9-bit binary weighted control signals. The signal CW[8:5] control the coarse cells and CW[4:0] control the fine cells. In BWDC, one path comprises of a fixed capacitance realized with the minimum-sized transistor and the other path comprises of a tuning capacitance that is realized by adjusting the size of transistor [23].

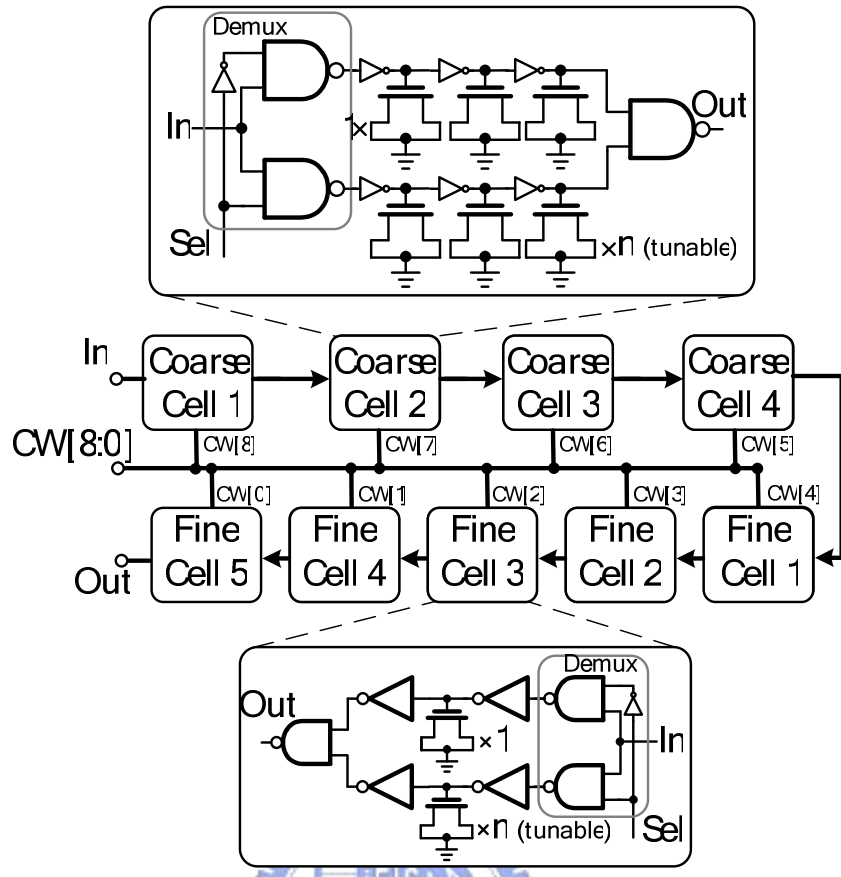


Figure 48 The architecture of DCDL and BWDC.

By the simulation results, the delay range can be selected 2ns to 10ns. Figure 49(a) shows simulation results of the delay time versus input control word. The power dissipation versus delay time is also shown in figure 49(b).

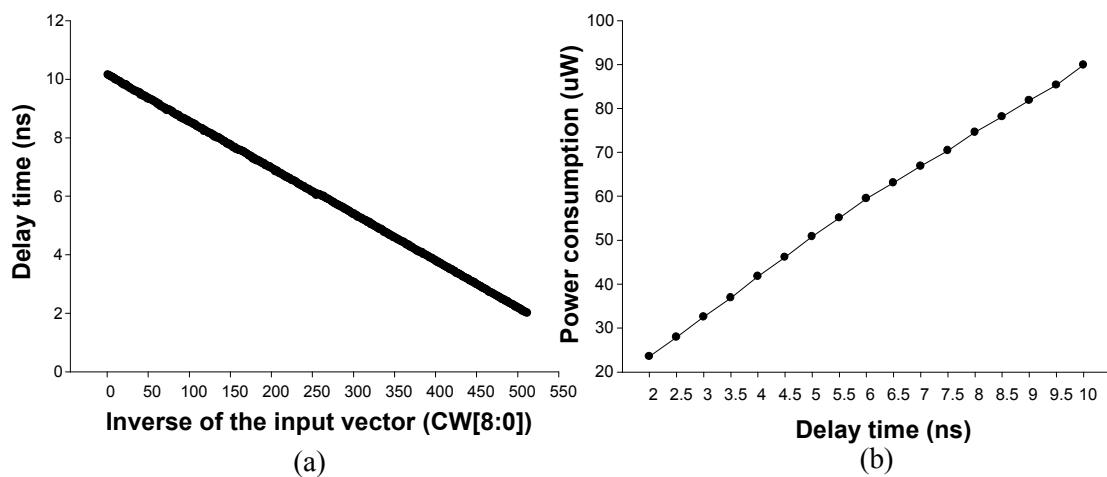


Figure 49 The simulation results of (a) delay time versus input vector (b) power consumption versus delay time.



## 4.5 SIMULATION RESULTS

The simulation waveforms of the unbalance binary search algorithm based DLL operates at 100MHz, 125MHz, 250MHz and 500MHz are shown in Figure 51, Figure 52, Figure 53, and Figure 54, respectively. The simulation environment is built in UMC 90nm CMOS technology with 1V supply voltage. The simulation result shows the DLL is functional within all the frequency. The locking produce takes 18 and 22 reference clock cycles for over 250MHz and below 250MHz, respectively. The Figure 51 consists of *Ref clock* signal, *Out clock* signal, *CLK* signal, *RN* signal, *LEAD* signal, *PS* signal, and the control word *CW[8:0]* signal. Those signals are stated as follows:

- (1)  $T_{REF}$ : The clock period of reference clock.
- (2) *Ref clock*: The external reference clock.
- (3) *Out clock*: The output of the DCDL.
- (4) *CLK*: The trigger signal for the UBS controller.
- (5) *RN*: Global reset signal.
- (6) *LEAD*: The output of phase detector. If *LEAD* is 1, it will increase the delay of DCDL otherwise it will decrease the delay of DCDL.
- (7) *PS*: Indicate whether  $T_{REF}$  is large twice of  $T_{DCDL\_MIN}$ .
- (8) *CW[8:0]*: The value of the DCDL control word register.
- (8) *STEP*: The content of the step controller.

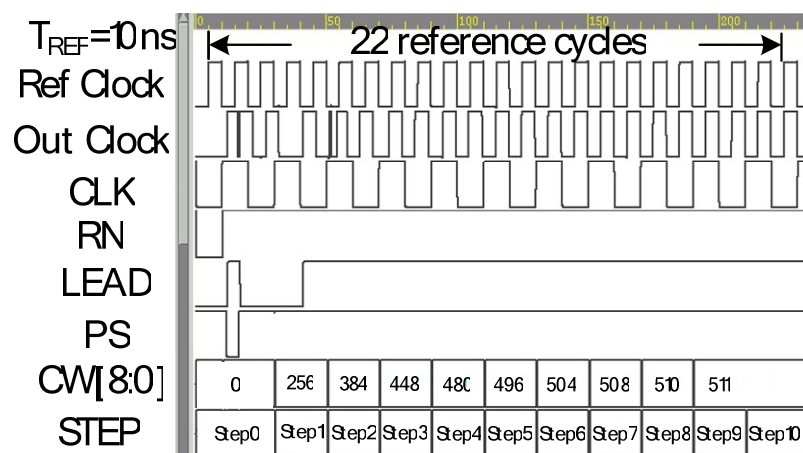


Figure 50 Lock process when the operating frequency at 100MHz

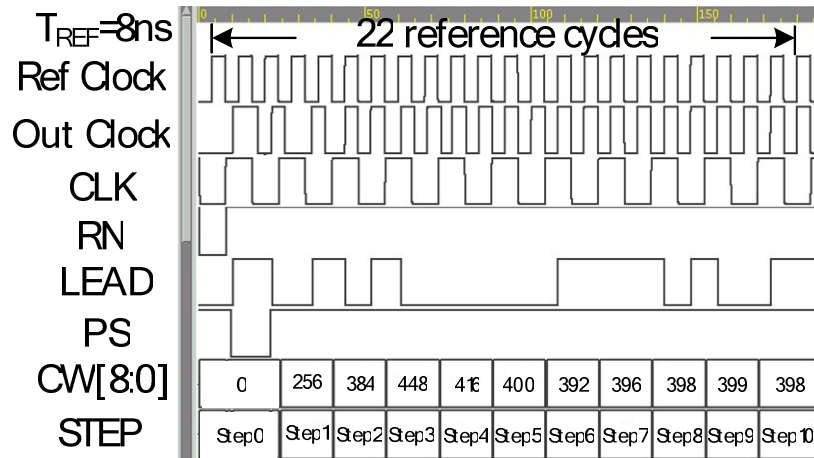


Figure 51 Lock process when the operating frequency at 125MHz

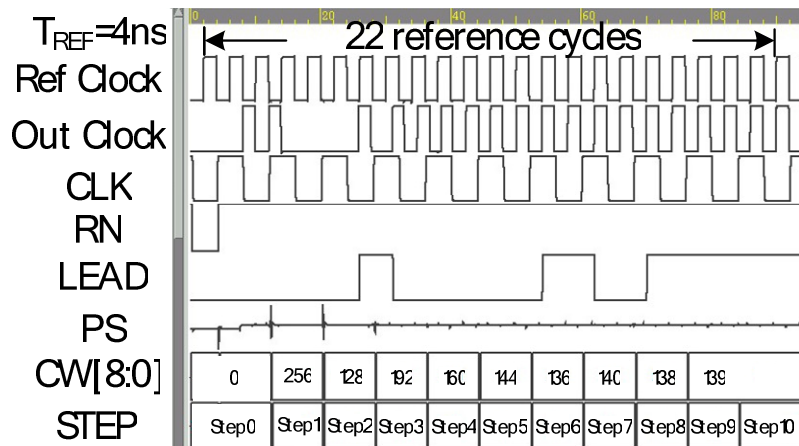


Figure 52 Lock process when the operating frequency at 250MHz

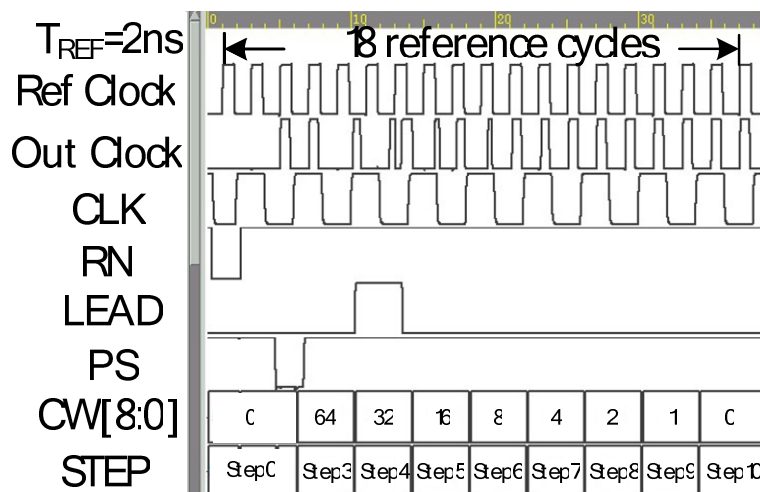


Figure 53 Lock process when the operating frequency at 500MHz

# CHAPTER 5

## IMPLEMENTATION OF ALL-DIGITAL FAST-LOCK SELF-CALIBRATED MULTIPHASE DLL

In this chapter, a 333MHz-1GHz all-digital multiphase delay-locked loop with precise multi-phase output has been designed with UMC 90nm CMOS technology model. A proposed UBS controller is used to match up a linear approximate delay element (LADE). The LADE property of linearity and insensitive to PVT variations is good for digitally-controlled delay element. The lock-in time could be reduced down to 14 reference clock cycles, and enhance the operation range based on LADE/UBS controller co-operate effort. The timing error caused by process mismatch is further reduced by proposed RSC algorithm. A calibration unit is designed based on RSC algorithm, which reduces the maximum timing error to less than 4.5ps when DLL is operating at 500MHz. The entire calibration unit could be turned off after calibration procedure is complete to reduce power consumption. The total power dissipation of the all-digital self-calibrated multiphase delay-locked loop is 2.16mW at 1GHz with a 1V power supply.

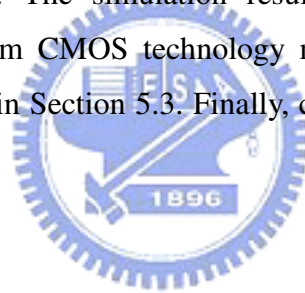
### 5.1 INTRODUCTION

Multi-phase delay-locked loop is a key component of various modern applications; these include dynamic frequency scaling, time-interleaved [2], and high frequency applications [18]. Traditionally, several identical delay stages are used to achieve multiphase generation in delay-locked loop. The identical delay stage is no longer “identical” after fabrication, not to mention temperature and supply voltage variations. On the other hand, analog DLL became not only difficult to design because

of process variation, but also easily interfered by noisy environment in mixed-signal SoC system. Therefore, a highly accurate multiphase output is required, calibration mechanism is essential for modern multiphase delay-lock loop. An all-digital multiphase delay-locked loop with rapid self-calibration ability is proposed.

An all-digital self-calibrated multiphase DLL (ADSCM-DLL) is proposed with rapid self-calibration ability. The unbalance binary search algorithm could accelerate the locking procedure and provide a wider locking range by setting an unbalanced control word. After the DLL is locked, self-calibration algorithm is initiated to compensate the phase error by adjusting the delay time of each delay stage. This could help the system to provide PVT variation immunity.

An overview of the proposed ADSCM-DLL is introduced in Section 5.1. Section 5.2 presents the circuit implementation to fulfill the intention of the proposed RSC algorithms which is presented in Section 3.4 and the proposed UBS algorithm which is presented in Section 4.3. The simulation results of ADSCM-DLL which is implemented in TSMC 130nm CMOS technology model and UMC 90nm CMOS technology model are shown in Section 5.3. Finally, conclusions are given in Section 5.4.



## 5.1 SYSTEM ARCHITECTURE

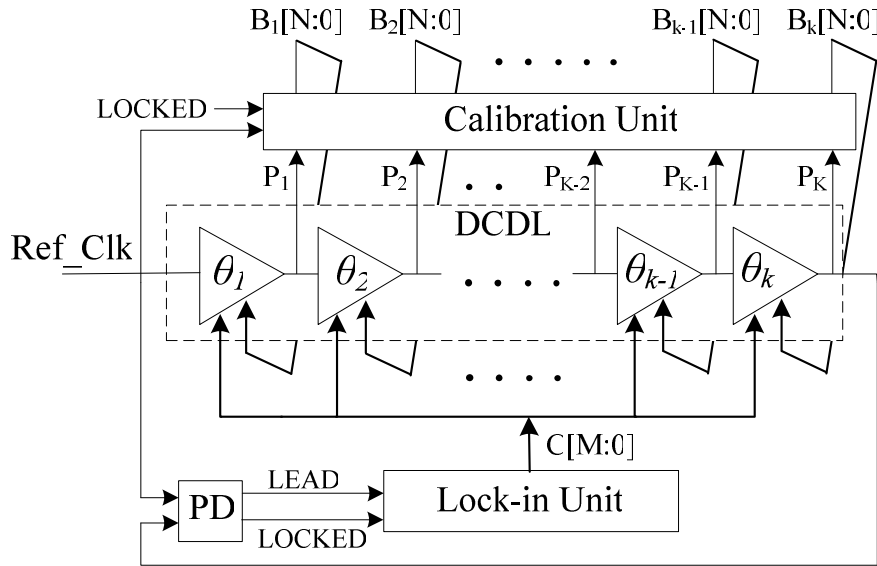


Figure 54 The proposed ADSCM-DLL architecture.

The proposed all-digital self-calibrated multiphase DLL, shown in Figure 1, consists of four major blocks; they are digitally controlled delay line (DCDL), phase detector (PD), lock-in unit, and calibration unit. There are  $K$  identical delay stages in the DCDL. All the delay stages are controlled by  $C[M:0]$  and  $B_i[N:0]$ .  $C[M:0]$  is generated by lock-in unit using the unbalance binary search algorithm. Meanwhile,  $B_i[N:0]$  is given by self-calibration unit, and it would carefully adjust the phase difference between every output signal after the DLL is locked.  $P_i$  stands for multiphase DLL output of  $i$ th delay stage.  $\theta_i$  symbolizes the phase difference between  $i$ th and  $(i-1)$ th stage of delay stage.

The lock-in sequence begins with comparing the phase difference between reference clock and  $P_k$ . With calibration unit disabled, the lock-in unit would adjust the control word  $C[M:0]$  based on unbalance binary search algorithm (Section 4.3). The total delay time of DCDL would be equal to one clock cycle of reference clock after binary search is completed. When (1) is satisfied, the *LOCKED* signal would be inserted. The calibration unit will perform output phase self-calibration by changing the control word  $B_i[N:0]$ , where as  $C[M:0]$  is fixed.

The operation of calibration unit is based on the proposed RSC algorithm as we

mentioned in Section 3.4. The calibration unit first considers about three signals; they are reference clock, P1, and P2. The digitally relative phase detector [7] is used to adjust  $\theta_1$  to  $(\theta_1+\theta_2)/2$  by changing the calibration control word, where  $\theta_i$  means the phase difference between  $P_i$  and  $P_{i-1}$ . Similarly, the calibration unit would consider about the next three signals; they are P1, P2, and P3. It will adjust  $\theta_2$  to  $(\theta_2+\theta_3)/2$  by changing the calibration control word. Unlike [7], the modified  $\theta_1$  does not affect  $\theta_2$ . This allows sequential adjustment in the same reference cycle. Finally, the adjustment of  $\theta_k$  is based on  $P_k$  and reference clock by the main loop, which guarantees the whole DLL remains locked. The control word  $C[M:0]$  remains unchanged during the calibration process to ensure successful RSC operation. As the result, the final output difference of each delay stage is one fifth of the period of reference clock.

## 5.2 CIRCUIT DESCRIPTION

### 5.2.1 Phase Detector

Low-jitter DLLs use commonly the linear phase detector (PD) or phase frequency detector (PFD) for the phase acquisition. However, when the clocking rate increases, the linear PD or PFD will limit the operating frequency of the DLL. Adding the dividers in front of the PD may mitigate this problem, however, the unequal delay time between the dividers will generate a significant static phase offset at the locked state. [7]

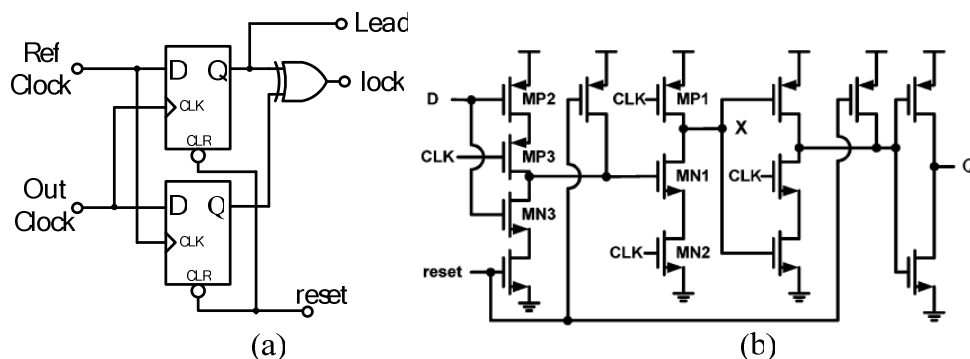


Figure 55 (a) The block diagram of phase detector (b) TSPC

The phase resolution is a key design parameter of the PD. The PD that consisted with two D\_Flip\_Flop(DFF) and some logic circuits[7] as shown in figure 55 (a) can achieve high resolution for the conventional all-digital DLL application. The PD examines whether reference clock leads output clock and generates output signals to drive next stage. When the reference clock and output clock are very close, the meta-stability of the DFF will lead the PD to output lock signals. In order to gain higher resolution the True Single-Phase Clocked (TSPC) based DFF is adopted [24] (Figure 55 (b)). Unfortunately, the floating output signals of the TSPC based PD cause a very long rise time (or fall time), that may make a violation for the next stage and not suitable for the high speed application. As figure 56 shows, when the reference clock leads the output clock around 4ps and operates at 2GHz. The output signal of PD, *Lead*, should down to zero. However, a long fall time makes a violation on the output node, *lock*.

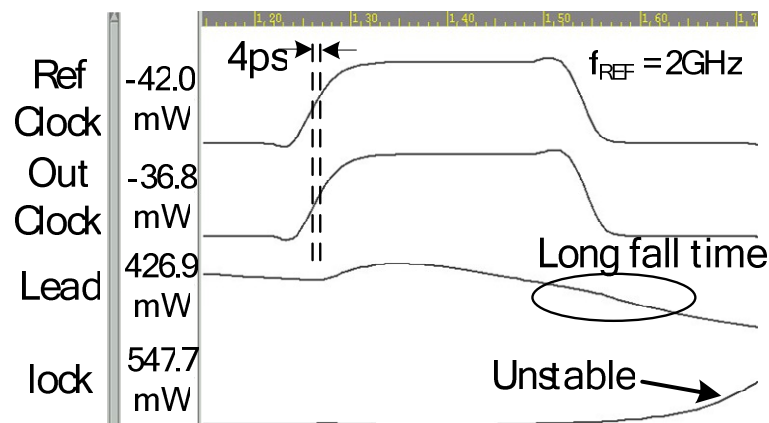


Figure 56 The operation of convention PD.

The proposed PD is based on the PD in [25] to take its advantages of high phase resolution. In order to avoid a long rise time (or fall time), a feedback circuit is added in the output of the TSPC based DFF as shown in Fig. 3(b). Therefore, the floating node can be eliminated. Figure 55 shows the simulation result for the proposed PD. Comparison with previous PD, the unstable state is vanished.

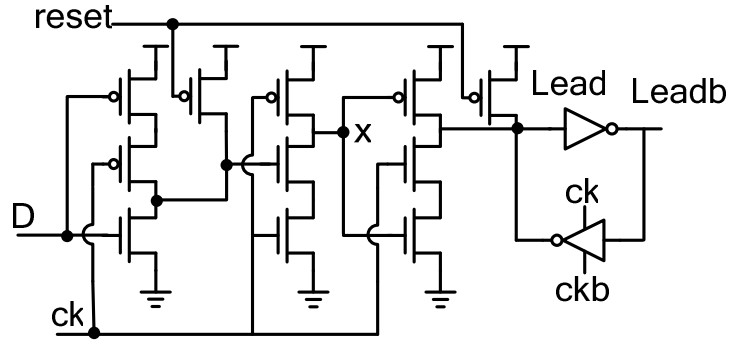


Figure 57 The modify TSPC DFF

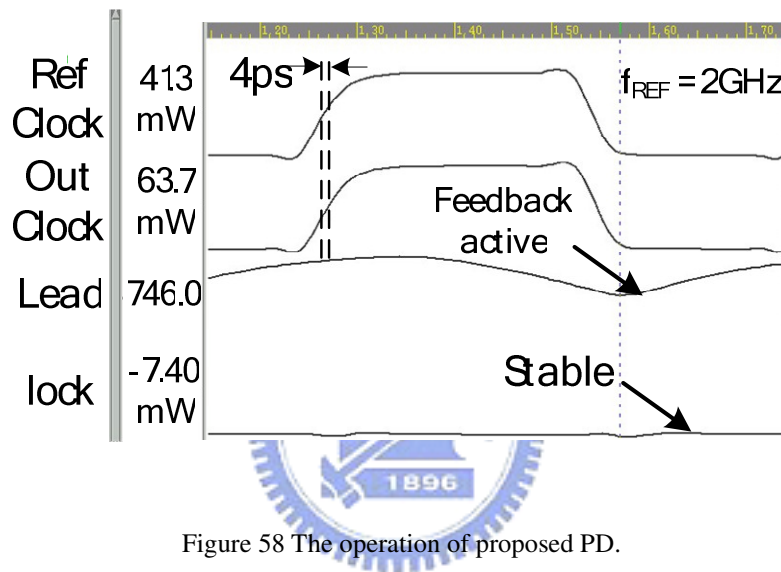


Figure 58 The operation of proposed PD.

## 5.2.2 LINEARLY APPROXIMANT DELAY ELEMENT

In order to gain higher PVT variation immunity, a current-starved-based digitally controlled delay element (DCDE) with strictly monotonic property has been proposed [8]. From [8], the empirical equation given as

$$t_d = \frac{A_1}{(A_2\sqrt{I_x} - A_3)} \quad (5.1)$$

Where,  $A_1$ ,  $A_2$  and  $A_3$  are constants. Current  $I_x$  dominates the delay time  $t_d$ . The empirical equation shows that the delay time is proportional to square root of current  $I_x$ . This property results in non-uniform delay increment when the control word vector value varies in large delay range. Therefore, delay time resolution is reduced when the current-starved-based DCDE operates in wide frequency range.



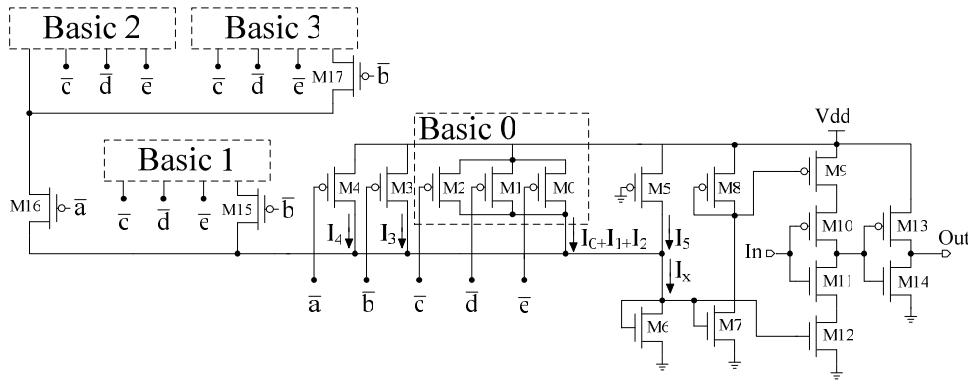


Figure 59 LADE

A novel linear-approximated delay element (LADE) is proposed to achieve an equal delay time increment property. The schematic view of LADE is shown in Figure 59. Each basic block is composed of three pMOS transistor, e.g. basic block 0 is composed of M0, M1, and M2. All blocks are controlled by the same control vector. Each basic block, however, has different transistor size to provide different current-driving strength. M15-M17 operate as switches to link each basic block to node x. This will accumulate all the current flowed from each basic block. Thus, the delay time is decided by the amount of  $I_x$ . The design procedure of LADE is shown as follows:

- I. The maximum delay time is decided by (W/L) ratio of M5. Then, adjusting the (W/L) ratio of M0-M3 for a proper delay increment.
- II. The transistors size of basic block 1 and 2 are identical ( $s_0=1$ ), which means two times of the original current will be provided to  $I_x$  when M3 is ON.
- III. Instead of setting the size of M4 to 2X of M3, 3X is needed to fit the linear increment principle because there is one more basic block as current source.
- IV. Finally, the size of basic block 2 and 3 are set to 2X ( $s_1=2$ ) and 4X ( $s_2=4$ ) of basic block 0.

$I_x$  can be expressed as (5.2) where  $I_5$  decides the maximum delay time that can be achieved by a LADE. In the previously discussed design procedure,  $I_0 = (1/2)*I_1 = (1/4)*I_2 = (1/8)*I_3 = (1/24)*I_4$  and  $1 = s_0 = (1/2)*s_1 = (1/4)*s_2$ . The value  $I_x$  is decided by linear programming.

$$I_x = I_5 + I_4 \cdot \bar{a} + I_3 \cdot \bar{b} + ((s_2 + s_1 \cdot \bar{b}) \cdot \bar{a} + s_0 \cdot \bar{b} + 1)(I_2 \cdot \bar{c} + I_1 \cdot \bar{d} + I_0 \cdot \bar{e}) \quad (5.2)$$

### 5.2.3 DIGITALLY CONTROLLED DELAY LINE

The Digitally Controlled Delay Line (DCDL) in the proposed architecture is composed of five stages of LADE. Each DCDL stage has two LADEs, which is shown in Figure 60. The Lock-in LADE is controlled by the lock-in unit to perform coarse tuning stage. The Calibration LADE is controlled by the Calibration unit to perform fine tuning stage and guarantees the phase difference of each adjacent output is equal to  $TREF/5$ .

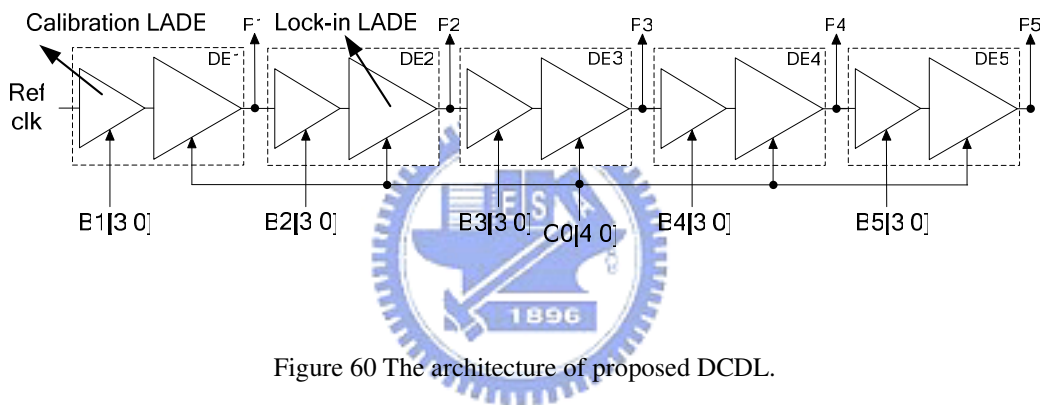


Figure 60 The architecture of proposed DCDL.

Five-phase output is supported by the proposed DLL. The characteristic curve of the five-stage DCDL is shown in Figure 61. The linearity of proposed LADE is much better than the current-starved-based delay element [8]. The average delay time step is 65ps.

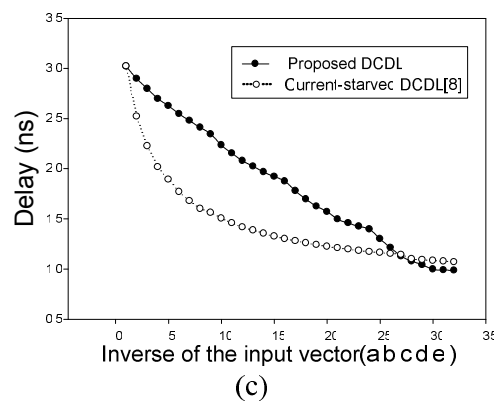


Figure 61 Delay of DCDL v.s. input vector

## 5.2.4 Lock-in unit

The architecture of lock-in unit is based on the UBS algorithm as we mentioned in Section 4.3. Figure 62 shows the 5-bit lock-unit which composed of the step controller and binary controller. By considering the characteristic of the DCDL, the skip steps of the lock-in unit only can achieve one. Therefore, the total cycles needed to complete a DLL lock based on the unbalance binary search algorithm are no more than 14 ( $7 \times 2$ ) reference cycles.

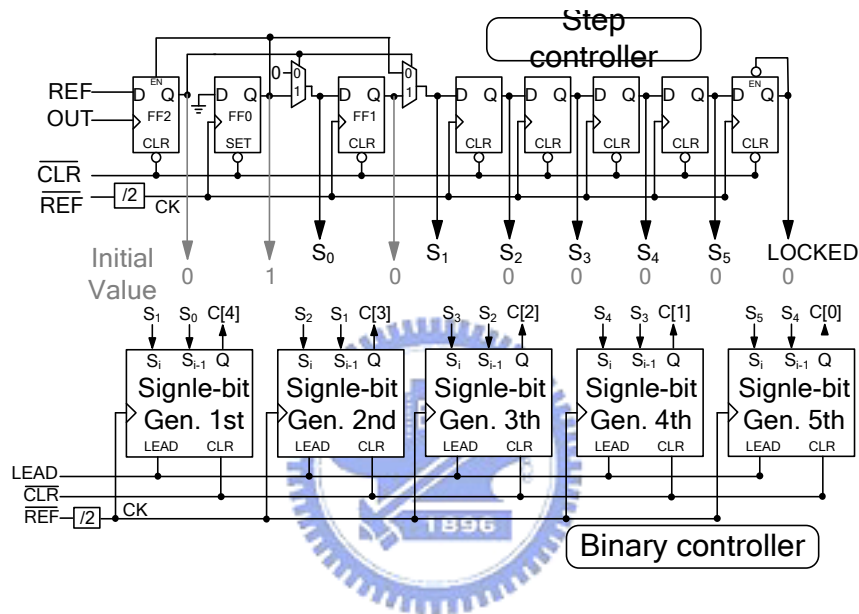
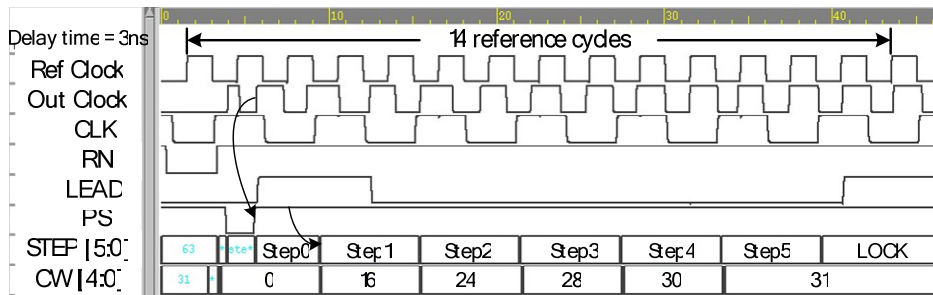
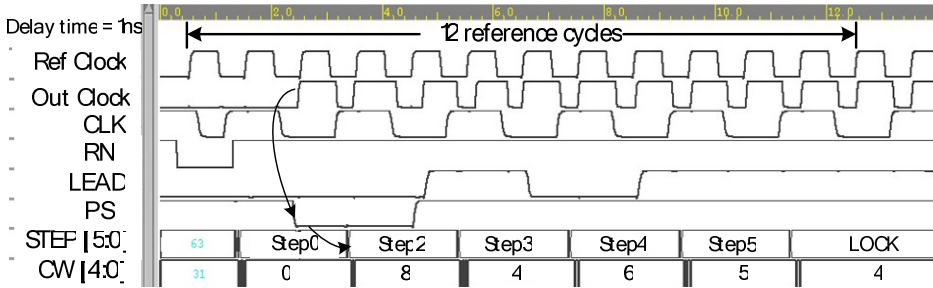


Figure 62 5-bit lock-in unit

Figure 63(a) shows the simulation result of the lock-in unit when the operating frequency is 333MHz. Assume the provided minimum delay of the delay line is 0.9ns and the  $T_{ref}$  is larger twice of  $T_{DCDL\_MIN}$ , therefore the lock time is 14 reference cycles. Similarly, the figure 63(b) shows the simulation result of the operating frequency is 1GHz, and the time is 12 reference cycles.



(a)



(b)

Figure 63 The operation of proposed lock-in unit

## 5.2.5 Calibration Unit

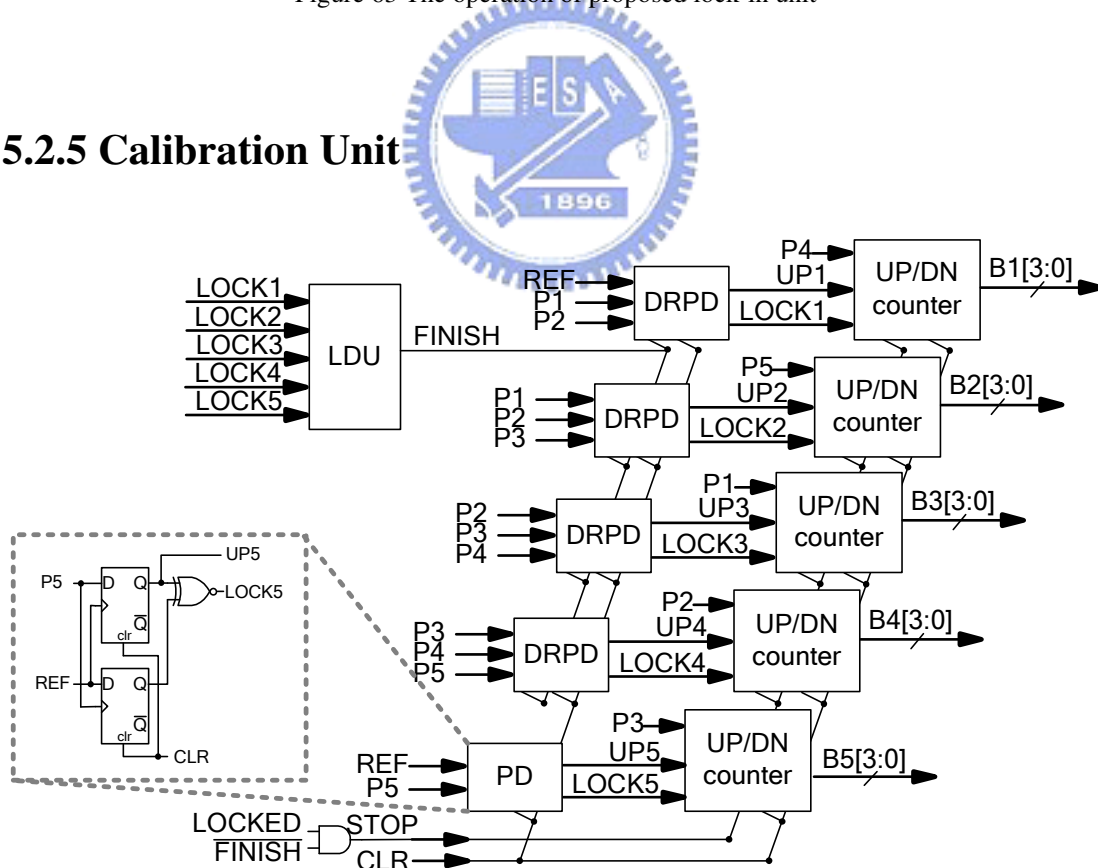


Figure 64 The block diagram of the calibration unit.

The calibration unit starts to function when *LOCKED* signal is inserted, which

compensates the phase difference mismatch caused by process variation. The schematic view of the proposed calibration unit is shown in Figure 64. It includes five up/down counters to control every single control word  $Bi[3:0]$  in  $i$ th delay stage. The lock detect unit (LDU) detects the lock state and generate a *FINISH* signal to disable the calibration unit. A conventional phase detector and four digitally relative phase detector (DRPD) [7] are also adopted to determine the relationship between several phase differences of input signals. The calibration unit continues to correct the phase mismatch until the relative phase error is smaller than the quantization error of DRPD. At this moment,  $LOCK_i$  signal is pulled up, and the  $i$ th control word  $Bi[3:0]$  value would be decided.

In our proposed circuit, self-calibration could continually adjust all five control words  $Bi[3:0]$  by updating the value of counters in the same reference cycle. Updated activity should be stable before the next positive edge of first reference signal. This requirement is achieved by using the far separated multiphase output as the trigger clock. No additional timing control circuit is needed. When all  $LOCK_i$  signals are low, it represents the relative phase error in each delay stage is smaller than quantization error. *FINISH* signal would be inserted to disable up/down counter and DRPDs. The control word  $Bi[3:0]$  would be fixed. The calibration unit would re-active when the relative phase error in some stage is bigger than quantization error of DRPD. This phase maintenance activity happens real-time and continuing.

In next section, we will describe the sub-circuits of proposed calibration unit in detail.

### 5.2.5.1 Digital Relative Phase Detector

In [3] the relative phase comparison method was proposed. Base on the method, the DRPD compares the relative phase error among the selected signals. The operation principle of DRPD is similar to [3] as shown in Figure 65. Since the proposed calibration circuit performs the phase comparison in the digital domain, the quantization error,  $q_e$ , is unavoidable.  $Pe(i)$  represents the phase error among  $(i-1)$ th,  $i$ th, and  $(i+1)$ th clock output. When the  $Pe(i)$  is out of the quantization error range, the DRPD generates the output an “up” or “down” signal to increase or decrease the delay

time of delay element  $i$ , respectively. If the  $Pe(i)$  is within the quantization error range, it will generate lock signal and not change the phase delay.

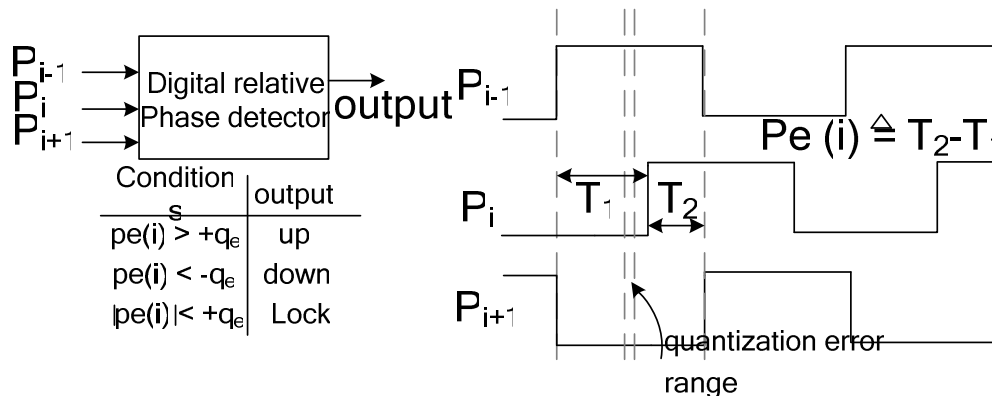


Figure 65 Operation of DRPD [3]

Figure 66 (a) shows the circuit design of DRPD which is composed of three proposed wide range interpolators, two DFFs and some logic circuits. The DRPD performs the relative phase comparison with using interpolators in the digital domain. Figure 66(b) shows the timing diagram of DRPD. The  $Sel$  signal is applied to interpolator for generate output signals with precisely weight factor of 50%. The homo-interpolating signal  $i22$  is generated by  $P2$  and itself. The hetero-interpolating signal  $i13$  is generated by  $P1$  and  $P2$ . If the homo signal  $i13$  leads hetero signal  $i22$ , i.e. the  $P2$  is under 50% of signal  $P1$  and  $P3$ . It will cause the output signal,  $UP$ , goes to high and update UP/DN counter to increase the delay time. When the  $i13$  is very close to  $i22$ , the unstable state may be occurred in the DFF and output wrong signals to UP/DN counter. In order to resolve this problem, an XOR gate is added to output a lock signal [7]. When the phase error is within the quantization error range, the signal  $lock$  will go high and freeze the state of the UP/DN counter. It ensures that the UP/DN counter will not be affected by the wrong updating signal from the PD. The simulation result shows the undetectable quantization error is around 7ps.

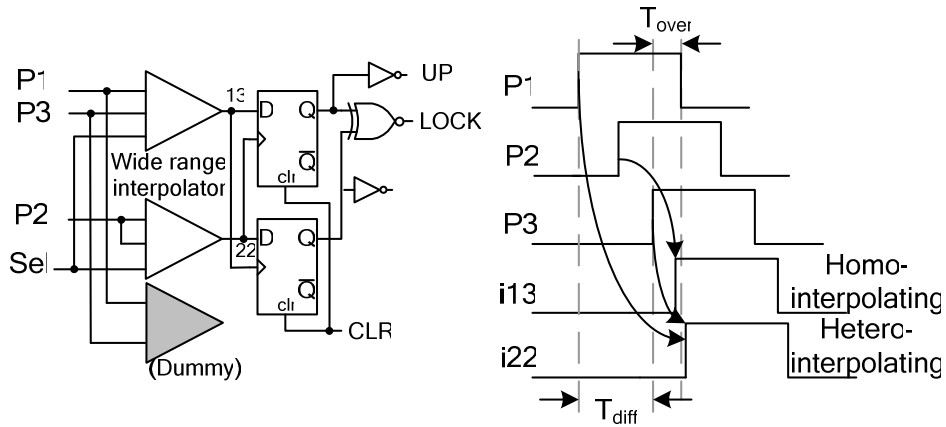


Figure 66 DRPD.

### 5.2.5.2 Interpolator

Since the interpolator is used to generate precise interpolation ratio, a short-circuit-current-suppression (SCCS) interpolator was proposed in [28]. By adding pre-charge circuit and some logic gates, it resolves the short-circuit-current between ground and supply voltage that causes the conventional digital interpolator can not output precise interpolation ratio.

The interpolation ratio and the operation ranges of the interpolator can be calculated with simple equations [28]. The delay time of a homo-interpolator and a hetero-interpolator are express as the following:

$$T_{\text{homo}} = C_{\text{th}} V_{\text{th}} / 2I \quad (5.3)$$

$$\begin{aligned} T_{\text{hetero}} &= T_{\text{diff}} + (C_{\text{th}} V_{\text{th}} - IT_{\text{diff}}) / 2I \\ &= T_{\text{homo}} + T_{\text{diff}} / 2 \end{aligned} \quad (5.4)$$

where  $T_{\text{diff}}$  is the time difference of two different phase signals. The  $V_{\text{th}}$  is the threshold voltage for inverting next gate.  $C_{\text{th}}$  is the capacitance for charging the next gate to  $V_{\text{th}}$ .  $I$  represents the current of the one of a parallel gate.

However, the operation range of the hetero-interpolator with precise interpolation ratio is restricted by equation (5.5).

$$0 < (C_{th} V_{th} - IT_{diff}) / 2I < T_{over} \quad (5.5)$$

where  $T_{over}$  is the overlap period of two different phase signals. When the operating frequency is increased  $K$  times, it means  $T_{over}$  and  $T_{diff}$  are also increase  $K$  times. However, the  $C_{th}$  is still fixed. Therefore, it may cause an imprecise interpolation ratio for the interpolator. The proposed wide-range interpolator is similar to [33] [34] and solves this problem by adding digitally controlled capacitors to extend the operation range. The digital control signals ( $S0$ ,  $S1$ ) are generated from the lock-in unit. Figure 67 shows the block diagram of the proposed wide-range interpolator.

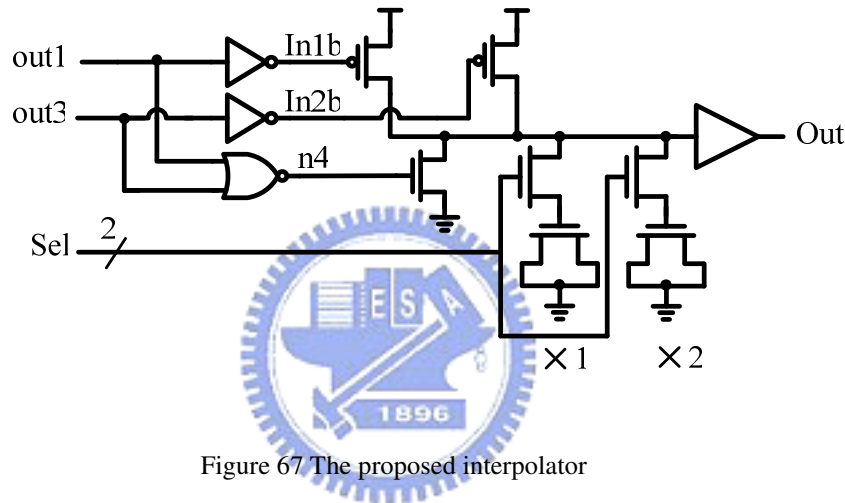


Figure 67 The proposed interpolator

### 5.2.5.3 Lock Detect Unit

When all the DRPD assert the  $LOCK_i$  signals, the relative phase error among all the output signals will be smaller than the quantization error. Thus, the calibration procedure is finished. Although the UP/DN counters will be freeze by the  $LOCK_i$  signal, the operation of interpolator is still working. Thus the additional power dissipation is consumed.

Figure 68 shows the architecture of the lock detect unit (LDU) which is composed of two D\_flip-flips and some logic gates. Once the signal  $LOCKED$  that form lock-in unit goes to high, the calibration enable signal,  $FINISH$ , will go high and start the calibration unit. When all the  $LOCK_i$  signal is asserted for two clock cycle, the LDU will active to disable all the calibrated circuit (including interpolator, phase



detector, up/down counter) to reduce power and the noise.

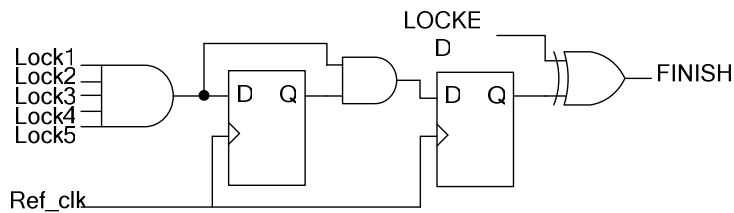


Figure 68 The lock detect unit

Figure 69 shows the simulation result for the proposed self-calibrated multiphase DLL with/without the LDU. The simulation results base on the UMC 90 CMOS technology and the operating frequency is 800MHz. When the proposed DLL is into the calibration stage, because of the operation of the interpolator and updating for the calibration unit result in the power consumption is increased. If the LDU is not applied to the calibration unit, even though there are no updating signals for the calibration unit, the power consumption is still high. Therefore, the LDU is needed. Comparison with the DLL without LDU, the DLL with LDU can save around 30% power consumption after the calibration is finished.

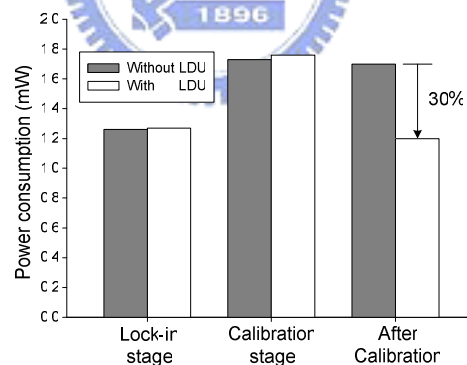


Figure 69 The power comparison of with/without LDU.

### 5.3 DESIGN IMPLEMENTATION

An all-digital self-calibrated multiphase DLL (ADSCM-DLL) is implemented in UMC 90 nm standard CMOS technology with 1V supply voltage. The major feature is the multiple outputs with precise phase. Therefore, it is important to make sure the loading seeming from each output node is the same through the layout consideration.

Besides, the delay from the reference signal and the final output of DCDL should be maintained to the same to ensure the correct locking detection. The proposed ADSCM-DLL can work correctly within +/- 10% voltage variation, 0°C to 100°C, and all the process corners. The output loading of the test chip is 0.5pF, which is the capacitance of the output pad. Therefore, output buffers are inserted to drive this large loading. The layout view of the ADSCM-DLL and the test chip are shown in Figure 70 and Figure 71, respectively.

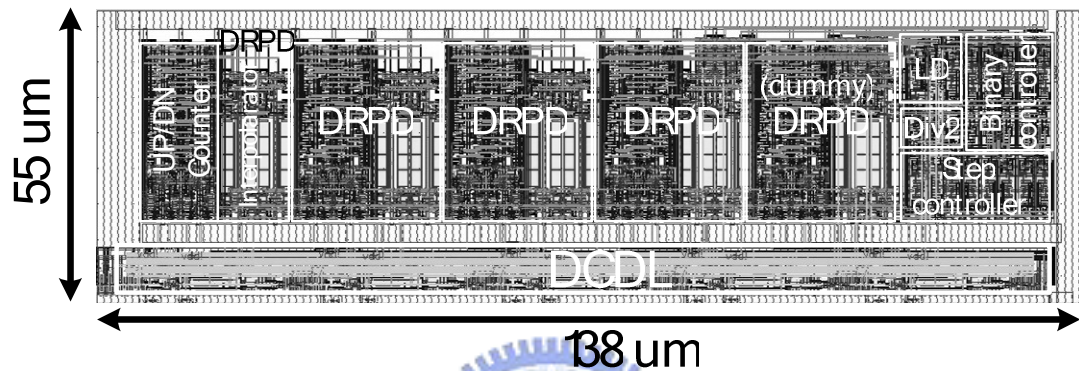


Figure 70 Layout view of the ADSCM-DLL

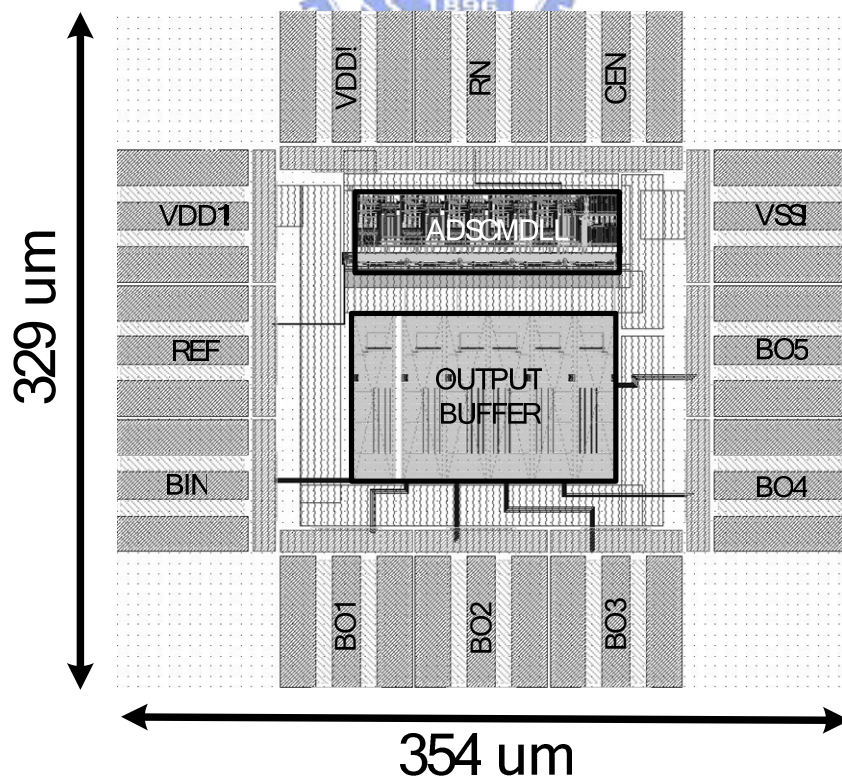


Figure 71 Layout view of the test chip

## 5.4 SIMULATION RESULT

The proposed all-digital self-calibrated multiphase DLL is simulated in UMC 90 nm standard CMOS technology. A. The operation frequency range is 300MHz – 1.08GHz. The locking procedure takes 12 and 14 reference clock cycles for over 480MHz and below 480MHz, respectively. The proposed delay-locked loop provides 8-bit resolution, and the LSB resolution is 4ps. The total power consumption is 1.1mW at 300MHz, and 2.16mW at 1.08GHz.

The process mismatch is provided by capacitance added within delay element. Figure 73 shows the waveform for the lock-in stage when the operating frequency is 500MHz. The signal *RN* and *RNI* reset the system, after that the lock-in unit base on the signal *LEAD* that is generated from PD to change the control word, *C[4:0]*, and traces the reference clock. The total locking cycles is 14 reference cycles and the signal *LOCKED* pulls up to start the calibration stage.

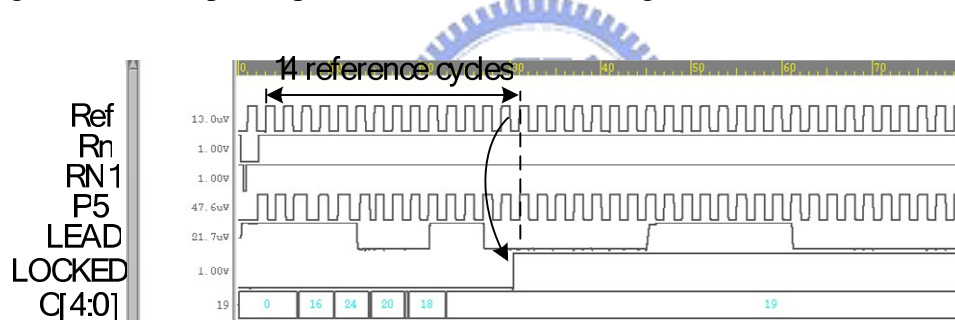


Figure 72 The operation of lock-in stage.

When the DLL is into the calibration stage, the calibration loops will adjust the relative phase error of output signals. The total calibration cycles is unpredictable, but in this case is 15 reference cycles. Figure 74 shows the waveform for the calibration stage.

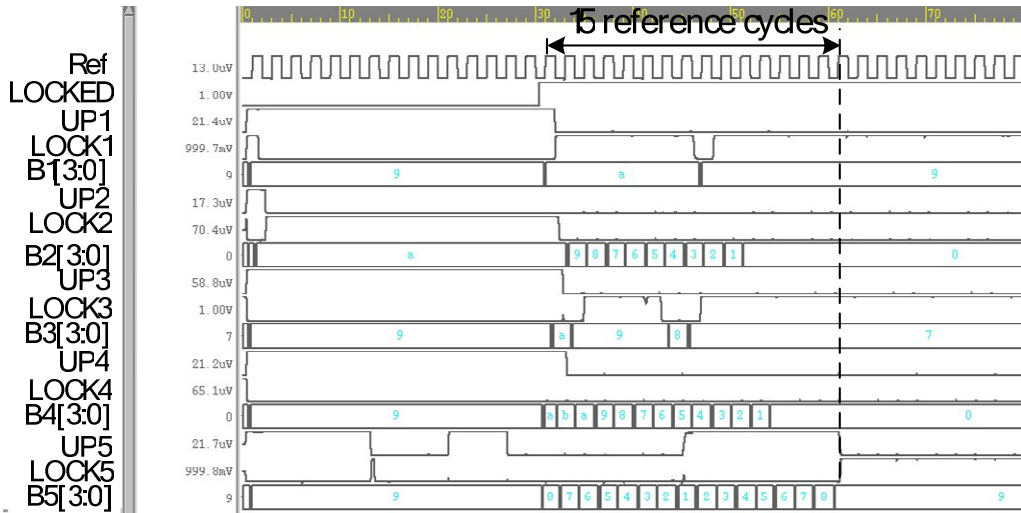
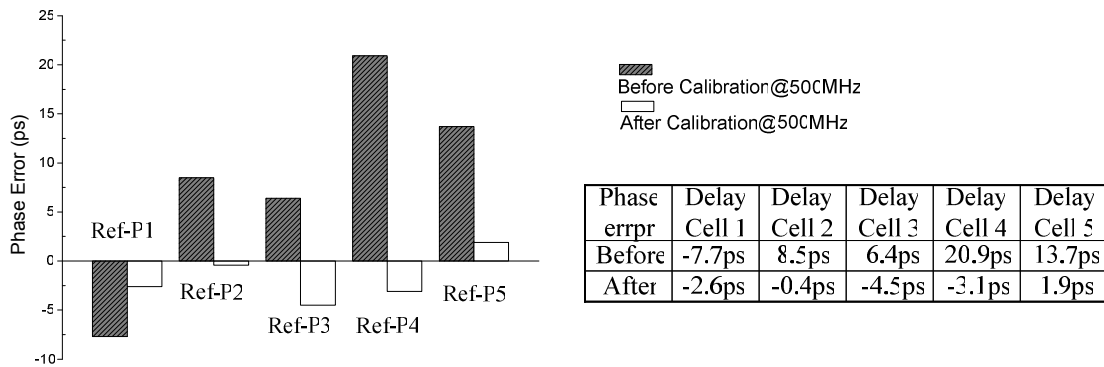
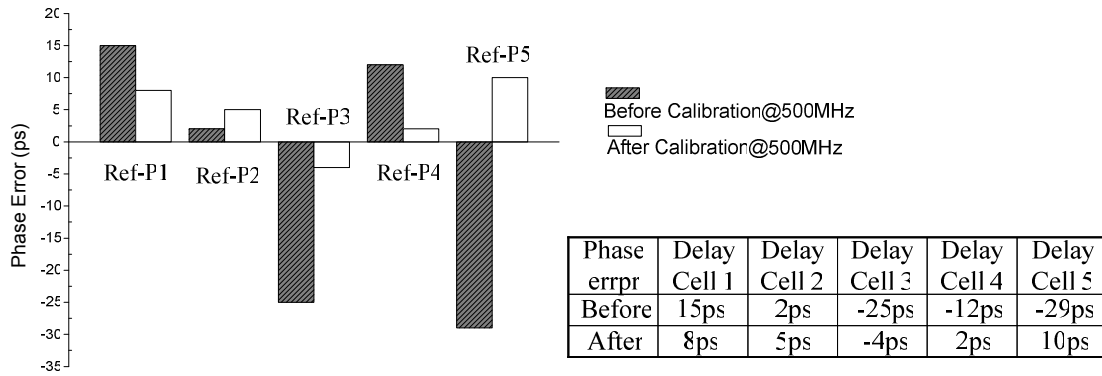


Figure 73 The operation of calibration stage

When the DLL operates at 500MHz, a maximum 20.9ps (or  $3.76^\circ = 20.9/2000 * 360$ ) phase error is occurred in one delay stage. After rapid self-calibration algorithm is applied, the maximum phase error could be reduced to 4.5ps (or  $0.81^\circ = 4.5/2000 * 360$ ). The phase error of each delay stage is shown in Figure 75(a). A comparison with TSMC 1.2V 130nm COMS technology model is also shown in Figure 75(b). The total simulation result of chip implementation summary and a comparison result under the UMC 90nm CMOS technology are shown in Table 3.



(a)



(b)

Figure 74 The phase error of each delay stage (a) 90nm (b) 130nm

Table 3 Summary of the ADSCM-DLL

All-Digital Self-Calibrated Multiphase Delay-Locked Loop		
Process	UMC 90nm	TSMC 130nm
Supply voltage	1V	1.2V
Operating frequency	300MHz-1.08GHz	333MHz- 1GHz
Effective sampling rate	1.5GHz- 5.4GHz	1.67GHz- 5GHz
Locking time	14 reference cycle	
Jitter (P-P)	@300MHz	13.4ps
	@1GHz	7.8ps
Before calibration		
Max. phase error	3.76°@500MHz	5.2°@500MHz
After calibration		
Max. phase error	0.81°@500MHz	1.8°@500MHz
Power	@1GHz	
	Core	2.16mW
	With Buffer	36.5mW
		N/A
@300MHz	Core	1.1mW
	With Buffer	18.5mW
Area	0.35X0.33mm <sup>2</sup>	N/A

The performance comparisons among previous works and this work are given in Table 4. Due to the finite digital quantization error, the calibrated phase error is larger than analog approach. However, this work proposed all-digital self-calibration architecture allows cost-effective integration with other circuits or portable with process.

Table 4 Comparison among previous works

Design	JSSC' 01 [25]	ISSCC' 01 [2]	ESSCC' 02 [3]	ISSCC'08 [42]	ISSCC'03 [27]	JSSC'06 [7]	This work
Process	0.35um	0.25um	0.18um	0.09um	0.35um	0.18um	0.09um
Supply voltage	3V	2.5V	2V	N/A	3.3V	1.8V	1V
Operating frequency	1.7-1.9 GHz	10-300 MHz	4-6 GHz	8-10 GHz	130-165 MHz	0.7-2 GHz	0.3-1.08 GHz
Output phases	4	8	4	5	10	5	5
Peak to Peak jitter	N/A	16.7ps @250MHz	N/A	2.04ps @10GHz	26.4ps @150MHz	18.9ps @2GHz	7.8ps @1GHz
DLL method	Analog	Analog	Analog	Analog	Analog	Analog	Digital
Calibration method	Analog	Analog	Analog	Analog	Analog	Digital	Digital
Before calibration Max. phase error	2.5° @1.8GHz	2.2° @125MHz	N/A	14.1° @9.5GHz	22.8° @150MHz	7.34° @1GHz	3.76° @500MHz
After calibration Max. phase error	0.2° @1.8GHz	0.32° @125MHz	2° @5GHz	4.8° @9.5GHz	6.1° @150MHz	1.26° @1GHz	0.81° @500MHz
Active area of Calibration unit	~0.16mm <sup>2</sup>	N/A	0.12mm <sup>2</sup>	~0	~0	0.52mm <sup>2</sup>	0.038mm <sup>2</sup>
Active area of whole chip	0.48mm <sup>2</sup>	1.6 mm <sup>2</sup>	45mm <sup>2</sup>	0.03mm <sup>2</sup>	2.25mm <sup>2</sup>	1.03mm <sup>2</sup>	0.116mm <sup>2</sup>
Power	60mW	80mW	2.7mW	15mW	18mW	81mW	36.5mW



# CHAPTER 6

## CONCLUSION AND FUTURE WORK

### 6.1 CONCLUSION

A novel 300MHz-1.08GHz all-digital self-calibrated multiphase delay-locked loop is proposed. An unbalance binary search algorithm extends the delay element operating frequency range by setting different initial delay time of delay line. Linear approximate delay element is implemented to harmonize with the modified binary search algorithm. The linear delay time increment is achieved, and solves the false lock problem. Meanwhile, the locking range is extended to entire delay line. Less than 14 reference clock cycles is needed for the locking procedure. A novel rapid self-calibration algorithm is also presented to overcome the delay element timing error caused by process variation. The calibration unit provides the maximum phase error to less than 4.5ps (0.81 degrees), which makes precise multiphase output with good PVT variation immunity possible. The proposed all-digital self-calibrated multiphase delay-locked loop is suitable for multi-core SoC applications.

### 6.2 FUTURE WORK

In the recent year, low power is more and more important issue in circuit design, especially for the portable devices which require ultra-low power consumption. Lowering supply voltage is a key focus in digital integrated circuit design [37]. Previous research demonstrates the functionality of logic circuits at 200mV using low threshold devices [38]. Furthermore, multiple independent clocks are in great demand for power management unit the multi-core systems [39]. Hence, in our future work, we look for a low power and multiple clock outputs with dynamic frequency scaling ability DLL based clock generated. We can take advantage of the multi-Vt CMOS (MTCMOS)

technique to increase the reliability while operating at low supply voltage such as 300mV. Beside, it also obtains low leakage and high speed at the same time.

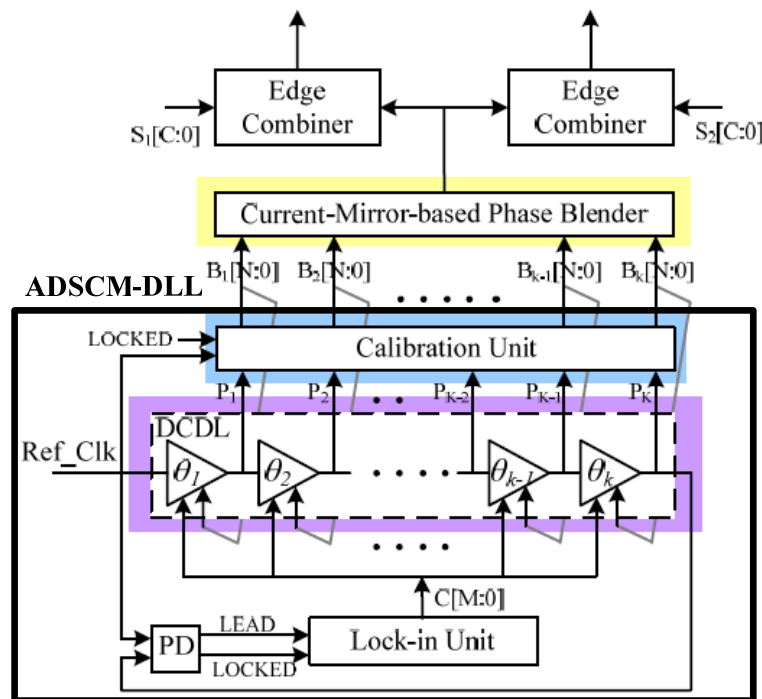


Figure 75 ADSCM-DLL based dual clock output generator

Figure 75 shows the idea of 300mV ADSCM-DLL based dual clock output generator. It includes an ADSCM-DLL, Current-Mirror-based Phase Blender (CMPB), and edge combiner (EC) [31]. The ADSCM-DLL reduces the mismatch effect and PVT variation, and then generates precise penta-phase outputs for the CMPB. It not only generates octal-phase signal from self-calibrated penta-phase signal, but also provides sufficient driving strength for EC. Each EC has independent programmability to choose desired frequency and phase by the control word  $S_i[C:0]$ . As this arrangement, the power-efficient multiphase dual clock output generator is suitable for autonomous long lifetime portable devices.



# BIBLIOGRAPHY

- [1] R. E. Best, *phase-Locked Loops: Theory, Design and Applications*, 2nd ed. New York: McGraw-Hill, 1993.
- [2] L. Wu and W. C. Black, Jr., "A low-jitter skew-calibrated multi-phase clock generator for tome-interleaved application," in *International Solid-State Circuits Conference*, Feb. 2001, pp. 396-399.
- [3] S. H. Wang et al., "A 5-GHz band I/Q clock generator using a self-calibration technique," *28th European Solid-State Circuits Conference*, Sep. 2002, pp. 807-810.
- [4] G. K. Dehng et al., "Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp.1128-1136, Aug. 2000.
- [5] R. J. Yang and S. I. Liu, "A 40-550 MHz Harmonic-Free All-Digital Delay-Locked Loop Using a Variable SAR Algorithm," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 361-373, Feb. 2007.
- [6] Y. Moon et al., "An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance," *IEEE J. Solid-State Circuits*, vol. 35, no. 3, pp. 377-384, Mar. 2000.
- [7] H. H. Chang et al., "A 0.7-2GHz Self-Calibration Multiphase Delay-Locked Loop," *IEEE J. Solid-State Circuits*, vol. 41, no. 5, pp.1051-1061, May 2006.
- [8] M. Maymandi-Nejad et al., "A Monotonic Digitally Controlled Delay Element," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2212-2218, Nov. 2000.
- [9] Kim, K.-h.; Chung, H.-J, " An 8 Gb/s/pin 9.6 ns Row-Cycle 288 Mb Deca-Data Rate SDRAM With an I/O Error Detection Scheme," *IEEE J. Solid State*

- Circuits, Vol. 42, issue 1, pp. 193-200 Jan. 2007.
- [10] Ching-Che Chung, Pao-Lung Chen, and Chen-Yi Lee, "An All-Digital Delay-Locked Loop for DDR SDRAM Controller Applications," VDAT Digital Object Identifier 10.1109 pp. 1-4, April, 2006.
- [11] K. H. Cheng, and Y. L. Lo, "A Fast-Lock Wide-Range Delay-Locked Loop Using Frequency-Range Selector for Multiphase Clock Generator," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 54, NO. 7, pp. 561-565, JULY 2007
- [12] H. H. Chang, J. W. Lin, C. Y. Yang, and S. I. Liu, "A Wide Range Delay Locked Loop with a Fixed Latency of One Clock Cycle," IEEE J. Solid-State Circuits, vol. 37, no. 8, pp. 1021-1027, Aug. 2002.
- [13] Jinn-Shyan Wang; Yi-Ming Wang; Chin-Hao Chen; Yu-Chia Liu;" An Ultra-Low-Power Fast-Lock-in Small-Jitter All-Digital DLL" in IEEE Intl. Solid-State Circuits Conference, Feb. 2005, pp. 422 – 607.
- [14] Hamamoto, T.; Furutani, K.; Kubo, T.; Kawasaki, S.; Iga, H.; Kono, T.; Konishi, Y.; Yoshihara, T.; "A 667-Mb/s operating digital DLL architecture for 512-Mb DDR SDRAM," IEEE J. Solid State Circuits, vol. 39, issue 1, pp. 194-206, Jan. 2004.
- [15] Y. J. Jeon, et al., "A 66-333MHz 12mW Register-Controlled DLL with a Single Delay Line and Adaptive Duty-Cycle Clock Dividers for Production DDR SDRAMs," IEEE J. Solid-State Circuits, vol. 39, no. 39, pp.2087-2092, Nov. 2004.
- [16] Tatsuya Matano et al., "A 1-Gb/s/pin 512-Mb DDRII SDRAM Using a Digital DLL and a Slew-Rate-Controlled Output Buffer," IEEE J. Solid-State Circuits, vol. 34, no. 5, pp.762-768, May. 2003.
- [17] Hsiang-Hui Chang, and Shen-Iuan Liu, "A Wide-Range and Fast-Locking All-Digital Cycle-Controlled Delay-Locked Loop," IEEE J. Solid-State Circuits, vol. 40, no. 3, pp.661-670, March, 2005.
- [18] Chi-Nan Chuang, and Shen-Iuan Liu, "A 0.5–5-GHz Wide-Range Multiphase

- DLL With a Calibrated Charge Pump,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, vol. 54, no. 11, pp.939-943, Nov, 2007.
- [19] Eunseok Song, Seung-Wook Lee, Jeong-Woo Lee, Joonbae Park, and Soo-Ik Chae, “A Reset-Free Anti-Harmonic Delay-Locked Loop Using a Cycle Period Detector,” IEEE J. Solid-State Circuits, vol. 39, no.11, pp.2055-2061, Nov. 2004.
- [20] Thomas Olsson, and Peter Nilsson, “A Digitally Controlled PLL for SoC Applications,” IEEE J. Solid-State Circuits, vol. 39, no.5, pp.751-760, May. 2004.
- [21] Yi-Ming Wang, and Jinn-Shyan Wang, “A Low-Power Half-Delay-Line Fast Skew-Compensation Circuit,” IEEE J. Solid-State Circuits, vol. 39, no.6, pp.906-918, June, 2004.
- [22] Rong-Jyi Yang, and Shen-Iuan Liu, “A 2.5 GHz All-Digital Delay-Locked Loop in 0.13 um CMOS Technology,” IEEE J. Solid-State Circuits, vol. 42, no.11, pp.2338-2347, Nov. 2007.
- [23] Jinn-Shyan Wang, Chun-Yuan Cheng, Yu-Chia Liu, and Yi-Ming Wang, “A 0.67mW/MHz, 5ps Jitter, 4 Locking Cycles, 65nm ADDLL,” ASSCC, Nov. 2007, pp.300-303.
- [24] A. Momtaz, C. Jun, M. Caresosa, A. Hairapetian, D. Chung, K. Vakilian, M. Green, T. Wee-Guan, J. Keh-Chee, I. Fujimori, and C. Yijun, “A fully integrated SONET OC-48 transceiver in standard CMOS,” IEEE J. Solid-State Circuits, vol. 36, pp. 1964–1973, Dec. 2001.
- [25] C. H. Park, O. Kim, and B. Kim, “A 1.8-GHz self-calibrated phase locked loop with precise I/Q matching,” IEEE J. Solid-State Circuits, vol. 36, pp. 777–783, May 2001.
- [26] F. Baronti, D. Lunardini, R. Roncella, and R. Saletti, “A self-calibrating delay-locked delay line with shunt-capacitor circuit scheme,” IEEE J. Solid-State Circuits, vol. 39, pp. 384–387, Feb. 2004.
- [27] H. H. Chang, C. H. Sun, and S. I. Liu, “A low jitter and precise multiphase

- delay-locked loop using shifted averaging VCDL,” in International Solid-State Circuits Conference, Feb. 2003, pp. 434–435.
- [28] T. Saeki, M. Mitsuishi, H. Iwaki, and M. Tagishi, “A 1.3-cycle Lock time, non-PLL/DLL clock multiplier based on direct clock cycle interpolation for “clock on demand”,” *IEEE J. Solid-State Circuits*, vol. 35, pp. 1581–1590, Nov. 2000.
- [29] Feng Lin, Jason Miller, Aaron Schoenfeld, Manny Ma, and R. Jacob Baker, “A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM,” *IEEE J. Solid-State Circuits*, vol. 34, no.4 pp. 565–569, April. 1999.
- [30] Byung-Guk Kim, and Lee-Sup Kim, “A 250-MHz–2-GHz Wide-Range Delay-Locked Loop,” *IEEE J. Solid-State Circuits*, vol. 40, no.6 pp. 1310–1321, June. 2005.
- [31] Jin-Han Kim, Young-Ho Kwak, Mooyoung Kim, Soo-Won Kim, and Chulwoo Kim, “A 120-MHz–1.8-GHz CMOS DLL-Based Clock Generator for Dynamic Frequency Scaling,” *IEEE J. Solid-State Circuits*, vol. 41, no.9 pp. 2077–2082, Sept. 2006.
- [32] Shao-Ku Kao, Bo-Jiun Chen, and Shen-Iuan Liu, “A 62.5–625-MHz Anti-Reset All-Digital Delay-Locked Loop,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS*, VOL. 54, NO. 7, pp. 566-570, JULY, 2007.
- [33] Stefanos Sidiropoulos, and Mark A. Horowitz, “A Semidigital Dual Delay-Locked Loop,” *IEEE J. Solid-State Circuits*, vol. 32, no.11 pp. 1683–1692, Nov. 1997.
- [34] Byung-Guk Kim, Lee-Sup Kim, Kwang Park, Young-Hyun Jun, and Soo-In Cho, “A DLL with Jitter-Reduction Techniques for DRAM Interfaces,” *International Solid-State Circuits Conference*, Feb, Feb. 2007, pp. 496-497.
- [35] Masakatsu Nakai et. al., “Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor,” *IEEE J. Solid-State Circuits*, vol. 40, no.1 pp. 28–35, Jan. 2005.

- [36] Toshihide Fujiyoshi et. al., "A 63-mW H.264/MPEG-4 Audio/Visual Codec LSI With Module-Wise Dynamic Voltage/Frequency Scaling," *IEEE J. Solid-State Circuits*, vol. 41, no.1 pp. 54–62, Jan. 2006.
- [37] B. H. Calhoun and A. Chandrakasan, "Ultra-dynamic voltage scaling using subthreshold operation and local voltage dithering in 90nm CMOS," in *International Solid-State Circuits Conference*, Feb. 2005, pp. 300-301, 599.
- [38] J. Burr and J. Shott, "A 200mV self-testing encoder/decoder using Stanford ultra-low-power CMOS," in *International Solid-State Circuits Conference*, Feb. 1994, pp. 84-85.
- [39] J. Dorsey, et al., "An integrated quad-core Opteron™ Processor," in *International Solid-State Circuits Conference*, Feb. 2007, pp. 102-103.
- [40] Kwangoh Kim, Nohman Park, and Taekyu Kim, "An Unlimited Lock Range DLL For Clock Generator," in *IEEE International Symposium on Circuits and Systems*, May 2004, pp. 776-779.
- [41] Jinn-Shyan Wang et al, "An Improved SAR Controller for DLL Applications," in *IEEE International Symposium on Circuits and Systems*, May 2006, pp. 3898-3901.
- [42] Keng-Jan Hsiao, and Tai-Cheng Lee, "A Low-Jitter 8-to-10GHz Distributed DLL for Multiple-Phase Clock Generation," in *International Solid-State Circuits Conference*, Feb. 2008, pp. 514-516.
- [43] G. Chien and P. Gray, "A 900 MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," in *IEEE International Solid-State Circuits Conference*, Feb. 2000, pp. 202–203.
- [44] C. Kim, I.-C. Hwang, and S.-M. Kang, "A low-power small-area 7.28-ps-jitter 1-GHz DLL-based clock generator," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1414–1420, Dec. 2002.
- [45] C.-K. K. Yang, R. Farjad-Rad, and M. A. Horowitz, "A 0.5-um CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling," *IEEE J. Solid-State Circuits*, vol. 33, pp. 713–721, May 1998.

- [46] Y. J. et al, "Synchronous Mirror Delay for Multiphase Locking," IEEE J. Solid-State Circuits, vol. 39, pp. 150–156, Jan. 2004.
- [47] D. S. et al, "An Analog Synchronous Mirror Delay for High-Speed DRAM Application," IEEE J. Solid-State Circuits, vol. 34, pp. 484–493, April 1999.
- [48] H. Sutoh, K. Yamakoshi, and M. Ino, "Circuit technique for skew-free clock distribution," in IEEE Custom Integrated Circuits Conf., 1995, pp. 163–166.
- [49] C. C. Chung, and C.Y. Lee, "A New DLL-Based Approach for All-Digital Multiphase Clock Generation," IEEE J. Solid-State Circuits, vol. 39, pp. 469–475, March. 2004.
- [50] J. S. Chiang and K. Y. Chen, "The design of an all-digital phase locked loop with small DCO hardware and fast phase lock," IEEE Trans. Circuits and System II, Analog Digit. Signal Process, vol. 46, no. 7, pp. 945–950, July 1999.
- [51] T. Y. Hsu and C. Y. Lee, "An All-Digital Phase-Locked Loop Based Clock Recover Circuit," IEEE J. Solid-State Circuits, vol. 34, pp. 1063–1073, Aug. 1999.
- [52] W. J. Choe et al, "A Single-Pair Link for Mobile Displays With Clock Edge Modulation Scheme," IEEE J. Solid-State Circuits, vol. 34, pp. 1063–1073, Aug. 1999.