國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

全數位寬電壓範圍寬頻率範圍延遲鎖定迴路時脈產生器設計

An All-Digital Wide Power Supply Range And Wide Frequency Range DLL-Based clock Generator

研 究 生：張益銘

指導教授：黃 威 教授

中 華 民 國 九 十 八 年 七 月

全數位寬電壓範圍寬頻率範圍延遲鎖定迴路時脈產生器設計

An All-Digital Wide Power Supply Range And Wide Frequency Range DLL-Based clock Generator

研 究 生：張益銘　　Student：Yi-Ming Chang

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 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

全數位寬電壓範圍寬頻率範圍延遲鎖定迴路時脈產生器設計

學生：張益銘　　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、寬電壓範圍、寬頻率範圍延遲鎖定迴路時脈產生器。藉由高穩健性的電路技巧使用，本電路能正常操作在寬電壓範圍。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，提出了一個可變性連續近似暫存器搜尋演算法，其特點在於結合頻率估計選擇器，提供不同的初始延遲時間已達到上述的功能。一個連續近似暫存控制器應用在多相位延遲鎖定迴路於 UMC 90nm CMOS 技術，模擬結果顯示，在 1.0V 工作電壓下，可操作頻率為 250MHz 到 1.25GHz ；在 0.3V 工作電壓下操作頻率為 13MHz 到 75MHz。

一個 125MHz 到 2.5GHz 全數位延遲鎖定迴路時脈產生器實現在 UMC 90nm CMOS 技術。為了達到動態調整頻率或電壓來減少功率消耗，一個穩健且可程式化的頻率倍乘器被提出。此外，一個新型的漏電流減少數位控制延遲元件被提出，根據模擬顯示，可減少 10%漏電流且不減緩電路的操作頻率。本時脈產生器其最大消耗的總功率為 0.71 毫瓦當操作在 500MHz 且產生 250MHz 、500MHz 、1GHz 及 2GHz 等四種不同頻率。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及可攜式產品中。

An All-Digital Wide Power Supply Range And Wide Frequency Range DLL-Based clock Generator

Student: Yi-Ming Chang Advisors: Prof. Wei Hwang

Department of Electronics Engineering & Institute of Electronics

National Chiao-Tung University

ABSTRACT

An all-digital wide power supply range wide frequency range DLL-based clock generator is proposed in this thesis. In order to operate in wide power supply range, the robust circuit methodology is used in this design. Besides, an adaptive successive approximation register-controlled (ASAR) search algorithm is proposed to extend the locking range and avoid harmonic lock at the same time. An ASAR based controlled is adopted in multiphase DLL implemented in UMC 90nm CMOS technology. The simulation results show that, the operating frequency is from 250MHz to 1.25GHz at 1.0V, and 13MHz to 75MHz at 0.3V, respectively.

A 125MHz-2.5GHz all-digital DLL-based clock generator has been designed in UMC 90nm CMOS technology. In order to achieve dynamic frequency/voltage scaling application, the robust and programmable frequency multiplier is proposed. Besides, the novel leakage-reduced delay unit is proposed to take advantages of mitigating 10% leakage current, insensitive to PVT variations, and not degrading operating frequency of circuit. The simulation results show the proposed DLL-based clock generator exhibits maximum power dissipation 0.71mW when operate in 500MHz, generating 250MHz, 500MHz, 1GHz, and 2GHz four different frequency at the same time. The presented DLL clock generator can be robustly used in embedded memory applications and portable device.

CONTENT

摘　　要……..……..I
ABSTRACT……..……II
CONTENT……..……III
LIST OF TABLES……..……..……...VI
LIST OF FIGURE……..……VII

CHAPTER 1 INTRODUCTION.....1

1.1 BACKGROUND 1

1.2 MOTIVATION 1

1.3 ORGANIZATION 2

CHAPTER 2 AN OVERVIEW OF DELAY-LOCKED LOOP.....4

2.1 THE ANALYSIS THEORY OF DELAY-LOCKED LOOP 4

2.2 DESIGN OF ANALOG DELAY-LOCKED LOOP 6

2.2.1 Stability Analysis of Delay-Locked Loop [2] 6

2.2.2 Jitter Analysis of Delay-Locked Loop [3] 8

2.2.3 Voltage-controlled Delay Line 9

2.2.4 Phase Detector 11

2.2.5 Charge Pump and Loop Filter 13

2.3 DIGITAL DELAY-LOCKED LOOP OVERVIEW 15

2.3.1 Register-controlled DLL [5] 16

2.3.2 Counter-controlled DLL [6] 17

2.3.3 Successive Approximation Register-controlled DLL [7] 18

2.3.4 Time measurement controlled DLL 20

2.4 CLASSIFICATIONS OF DIGITALLY CONTROLLED DELAY LINE 22

2.4.1 Shunt Capacitor based DCDE 22

2.4.2 Standard Cell based DCDE 23

2.4.3 Low power DCDE 26

2.4.4 Current-starved based DCDE 26

2.5 COMPARISON OF DIFFERENT TYPES OF DLLS 29

CHAPTER 3 MULTIPHASE DLL AND DLL-BASED FREQUENCY MULTIPLIER 30

3.1 INTRODUCTION OF MULTIPHASE DLL 30

3.2 APPLICATION OF MULTIPHASE DLL 30

3.2.1 DDR SDRAM Controller Application [17] 30

3.2.2 Transmitter [9] *33*

3.2.3 High-performance Microprocessors [18] *34*

3.3 INTRODUCTION OF DLL-BASED FREQUENCY MULTIPLIE 35

3.3.1 Basic of the DLL-Based Frequency Multiplier *35*

3.3.2 Timing Jitter Accumulation [19] *36*

3.4 APPLICATION OF DLL-BASED FREQUENCY MULTIPLIE 38

3.4.1 Local Oscillator for PCS Application [20] *38*

3.4.2 A Multiplying DLL for High Speed On-chip Clock Generation [21] *40*

3.4.3 Clock Generator for Dynamic Frequency Scaling *41*

CHAPTER 4 A WIDE POWER SUPPLY RANGE, WIDE LOCKING RANG ALL-DIGITAL MULTIPHASE DLL 44

4.1 INTRODUCTION OF WIDE-RANGE DLL 44

4.2 PREVIOUS RESEARCH OF WIDE-RANGE DLL 46

4.3 ADAPTIVE SAR ALGORITHM 50

4.4 CIRCUIT DESCRIPTION 52

4.4.1 Frequency-estimation Selector *52*

4.4.2 Adaptive SAR Controller *54*

4.4.3 Digitally Controlled Delay Line *55*

4.4.4 A Robust Ultra-Low Power Design *56*

4.5 SIMULATION RESULTS 60

CHAPTER 5 IMPLEMENTATION OF ALL-DIGITAL PROGRAMMABLE DLL-BASED FREQUENCY MULTIPLIER 63

5.1 SYSTEM ARCHITECTURE 64

5.2 CIRCUIT DESCRIPTION 65

5.2.1 Phase Detector *65*

5.2.2 Leakage-reduced Delay Line *66*

5.2.3 Lock-in Unit *72*

5.2.4 A Robust, Low Power, And Duty Cycle Immunity Frequency Multiplier *74*

5.2.5 Dynamic Frequency Monitor Circuit *78*

5.3 DESIGN IMPLEMENTATION 80

5.4 SIMULATION RESULT 81

CHAPTER 6 CONCLUSION AND FUTURE WORK.....83

6.1 CONCLUSION 83

6.2 FUTURE WORK 83
BIBLIOGRAPHY 85

LIST OF TABLES

TABLE 1 COMPARISON OF DIFFERENT TYPE DCDE 28

TABLE 2 COMPARISON OF DIFFERENT TYPES DLL 29

TABLE 3 COMPARISON AMONG PREVIOUS WORKS 82

LIST OF FIGURE

FIGURE 2.1 BLOCK DIAGRAM OF CONVENTIONAL DLL 4
FIGURE 2.2 BLOCK DIAGRAM OF ANALOG DLL 6
FIGURE 2.3 SMALL SIGNAL AC MODEL OF THE ANALOG DLL 6
FIGURE 2.4 LONG-TERM JITTER AND CYCLE-TO-CYCLE JITTER 8
FIGURE 2.5 THE CSDL (A) DELAY ELEMENT (B) DELAY LINE 9
FIGURE 2.6 THE RCDL (A) DELAY ELEMENT (B) DELAY LINE 10
FIGURE 2.7 THE DVCDL (A) DELAY ELEMENT (B) DELAY LINE 10
FIGURE 2.8: THREE-STATE PHASE DETECTOR 11
FIGURE 2.9: PD RESPONSES WITH (A) REFERENCE SIGNAL LAG FEEDBACK SIGNAL (B) REFERENCE SIGNAL LEAD FEEDBACK 12
FIGURE 2.10: (A) THE HALF-TRANSPARENT REGISTER (B) DYNAMIC PD 12
FIGURE 2.11: PD STATE DIAGRAM 13
FIGURE 2.12: SIMPLE MODEL OF CHARGE PUMP AND LOOP FILTER 14
FIGURE 2.13: THE SIMPLIFIED TIMING DIAGRAM OF CHARGE PUMP 14
FIGURE 2.14 LOOP FILTER 15
FIGURE 2.15 BLOCK DIAGRAM OF DIGITAL DLL 15
FIGURE 2.16 REGISTER CONTROLLED DLL 16
FIGURE 2.17 COUNTER-CONTROLLED DLL 17
FIGURE 2.18 SARDLL 18
FIGURE 2.19 FLOWCHART OF 3-BIT BINARY SEARCH ALGORITHM 19
FIGURE 2.20 TMDLL 20
FIGURE 2.21TDC CIRCUIT 21
FIGURE 2.22 DCDL REALIZED BY A PATH-SELECTION METHOD 22
FIGURE 2.23 SHUNT-CAPACITOR BASED DCDE 23
FIGURE 2.24 PARALLEL TRI-STATE INVERTER BASED DCDE 24
FIGURE 2.25 AOI-OAI PARALLEL BASED DCDE 24
FIGURE 2.26 (A) CIRCUIT WITH DIGITAL CONTROL. (B) EQUIVALENT CIRCUIT WITH _C CAPACITANCE. 25
FIGURE 2.27 COMPARISONS AMONG THE PROPOSED DCV AND OTHER APPROACHES 25
FIGURE 2.28: THE BINARY-WEIGHTED DIFFERENTIAL-DELAY CELL. 26
FIGURE 2.29 CURRENT STARVED BASED DCDE 27
FIGURE 3.1 READ OPERATION TIMING BUDGET 31
FIGURE 3.2 ARCHITECTURE OF MULTIPHASE DLL FOR DDR SDRAM APPLICATION 32
FIGURE 3.3 7:1 DATA CHANNEL COMPRESSION TRANSCEIVER. (A) TRANSMITTER CIRCUIT. (B) RECEIVER CIRCUIT. 33
FIGURE 3.4 ORGANIZATION OF THE MULTIPHASE CLOCK GENERATOR. 34
FIGURE 3.5 ORGANIZATION OF THE MULTIPHASE CLOCK GENERATOR. 36
FIGURE 3.6: OPERATION TIMING DIAGRAM OF FREQUENCY MULTIPLIER 36
FIGURE 3.7 TIMING JITTER ACCUMULATION FOR RING OSCILLATOR AND DELAY CHAIN. 37
FIGURE 3.8 BLOCK DIAGRAM FOR THE EXPERIMENTAL PROTOTYPE 39

FIGURE 3.9 EDGE COMBINER SCHEMATIC 39
FIGURE 3.10 BASIC COMPONENTS OF A HIGH-SPEED SERIAL I/O 41
FIGURE 3.11 MULTIPLYING DLL 41
FIGURE 3.12 COMPARISON OF POWER CONSUMPTION 42
FIGURE 3.13 BLOCK DIAGRAM OF DLL-BASED CLOCK GENERATOR 43
FIGURE 4.1 HARMONIC LOCKING PROBLEMS. 45
FIGURE 4.2 CONFIGURATION AND OPERATION OF A REPLICA DELAY LINE. 46
FIGURE 4.3 (A) LOCK-DETECT DECODE CIRCUITRY (B) PHASE DETECTOR SCHEMATIC. 47
FIGURE 4.4 NINE-STAGE VCDL WAVEFORMS WITH (A) CORRECT LOCK AND (B) FALSE LOCK 47
FIGURE 4.5 THE TDC ARCHITECTURE 48
FIGURE 4.6 (A) THE FLOWCHART OF THE VSAR ALGORITHM (B) VSAR CONTROLLER ... 49
FIGURE 4.7 SIMULATED LOCK TIME VERSUS THE OPERATING FREQUENCY 49
FIGURE 4.8: THE LOCKING RANGE PRINCIPLE OF THE PROPOSED ADAPTIVE SAR. 51
FIGURE 4.9 THE FREQUENCY-ESTIMATION SELECTOR 53
FIGURE 4.10: THE FREQUENCY-ESTIMATION SELECTOR TIMING DIAGRAM 53
FIGURE 4.11 7-BIT ADAPTIVE SAR CONTROLLER 54
FIGURE 4.12 LATTICE DELAY UNITS 55
FIGURE 4.13 FINE TUNE DELAY CELLS 55
FIGURE 4.14 POWERPC MASTER-SLAVER LATCH (POWERPC). 56
FIGURE 4.15 MODIFIED C2MOS MASTER-SLAVER LATCH (MC2MOS). 57
FIGURE 4.16 HYBRID-LATCH (HLFF). 57
FIGURE 4.17 SENSE-AMPLIFIER-BASED (SAFF) 57
FIGURE 4.18 TIMING PARAMETERS OF THE FLIP-FLOP AS A FUNCTION OF THE SUPPLY VOLTAGE. 58
FIGURE 4.19 ENERGY DISSIPATION AS A FUNCTION OF THE SUPPLY VOLTAGE FOR DIFFERENT SWITCHING 59
FIGURE 4.20 EDP AS A FUNCTION OF SUPPLY VOLTAGE AND SWITCHING ACTIVITIES. 59
FIGURE 4.21 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 250MHZ AT1V (S1S0=11) 61
FIGURE 4.22 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 667MHZ AT1V (S1S0=01) 61
FIGURE 4.23 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 1GHZ AT1V (S1S0=00) 61
FIGURE 4.24 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 18 MHZ AT 0.3V (S1S0=11) 62
FIGURE 4.25 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 40 MHZ AT 0.3V (S1S0=01) 62
FIGURE 4.26 LOCK PROCESS WHEN THE INPUT FREQUENCY IS 67 MHZ AT 0.3V (S1S0=00) 62
FIGURE 5.1: THE PROPOSED ALL-DIGITAL DLL-BASED FREQUENCY MULTIPLIER ARCHITECTURE 64
FIGURE 5.2 (A) THE BLOCK DIAGRAM OF PHASE DETECTOR (B) POWERPC FLIP-FLOP ... 66
FIGURE 5.3 OPERATION PRINCIPLE OF PROPOSED PD 66
FIGURE 5.4 LEAKAGE CURRENT COMPONENTS IN AN NMOS TRANSISTOR 67
FIGURE 5.5 TWO-INPUT NAND GATE STACKING EFFECT ILLUSTRATION. 68

FIGURE 5.6: INPUT VECTOR VERSUS LEAKAGE CURRENT. 69
FIGURE 5.7: (A) NAND (B) HALF-STACK NAND 69
FIGURE 5.8 STANDBY POWER COMPARISONS 70
FIGURE 5.9 ACTIVE POWER COMPARISONS 70
FIGURE 5.10 TIME DELAY COMPARISONS 70
FIGURE 5.11: THE PROPOSED LEAKAGE-REDUCED DELAY LINE 72
FIGURE 5.12: STANDBY POWER COMPARISONS 72
FIGURE 5.13: TIME DELAY COMPARISONS 72
FIGURE 5.14: 7-BIT LOCK-IN UNIT 73
FIGURE 5.15: THE OPERATION OF PROPOSED LOCK-IN UNIT AT 1GHZ 74
FIGURE 5.16: 05X FREQUENCY GENERATION CIRCUIT 75
FIGURE 5.17: 1X FREQUENCY GENERATION CIRCUIT 75
FIGURE 5.18: OUT_1X OF INPUT CLOCK FREQUENCY WHEN INPUT (A) 25% DUTY CYCLE (B) 75% DUTY CYCLE 76
FIGURE 5.19: 2X FREQUENCY GENERATION CIRCUIT 76
FIGURE 5.20: OUT_2X OF INPUT CLOCK FREQUENCY WHEN INPUT (A) 25% DUTY CYCLE (B) 75% DUTY CYCLE 77
FIGURE 5.21: 4X FREQUENCY GENERATION CIRCUIT 78
FIGURE 5.22: TIMING DIAGRAM 4X FREQUENCY GENERATION CIRCUIT 78
FIGURE 5.23: THE DYNAMIC FREQUENCY-MONITOR CIRCUIT 79
FIGURE 5.24: TIMING DIAGRAM OF THE DFM 79
FIGURE 5.25: LAYOUT VIEW OF THE ADP-DLLFM 80
FIGURE 5.26: LAYOUT VIEW OF THE TEST CHIP 80
FIGURE 5.27: GENERATE FOUR DIFFERENT FREQUENCY WITH 1.0V POWER SUPPLY 81
FIGURE 5.28: GENERATE FOUR DIFFERENT FREQUENCY WITH 0.3V POWER SUPPLY 81
FIGURE 6.1 ADCGSC CLOCK GENERATOR 84

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

With the evolution 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. Recently, the DLL clock generators draw more attention due to some better innate characteristic than PLLs, such as simple design effort and stable to PVT variations. 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 DLL-based clock generator is used in many high performance applications, such as clock/data recovery (CDR) circuit [33], double data rate (DDR) SDRAM [5] [17], and frequency multiplier [20] [21] [23] [35]. A multiphase VCDL or DCDL output is typically used to implement this circuit function. The conventional

multiphase 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. For example, an all-analog DLL [6] improves the locking range by using replica delay line. However, it is not suitable for the process portability and noise immunity consideration. The frequency multiplier is one of applications of DLL, and it is usually a key component of local oscillator and clock generator. However, conventional frequency multipliers dissipate large power [20] [21] and may have serious malfunction in wide supply voltage range [23][34].

According to above issues, this thesis focuses on the techniques of the search algorithm for the DLL to eliminate false locking problem; in addition, the programmable and robust frequency multipliers is proposed for the dynamic frequency/voltage scaling application.

1.3 ORGANIZATION

The thesis includes six chapters which focus on wide power supply range, wide locking range DLL-based clock generator. Following briefly introduces the content of each chapter.

Chapter 2 gives an overview of DLL, including analog DLL and digital DLL. comparison results are also given in this chapter.

Chapter 3 describes the fundamentals and applications of multiphase DLLs and frequency multipliers.

Chapter 4 introduces the proposed adaptive successive approximation register-controlled search algorithm which avoids harmonic locking in wide-locking range. Simultaneously, a robust and ultra-low voltage design methodology is discussed.

Chap 5 presents a programmable DLL-based clock generator. Base on the

proposed frequency multiplier, the clock generator is suitable for dynamic frequency/voltage scaling which low power application is required. 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

2.1 THE ANALYSIS THEORY OF DELAY-LOCKED LOOP

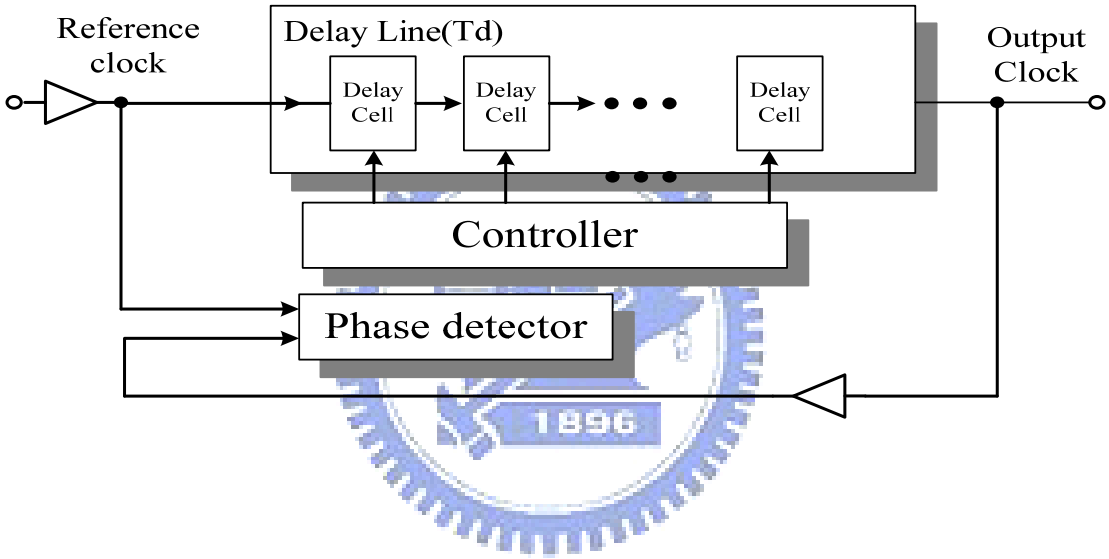


Figure 2.1 Block diagram of conventional DLL

A simplified block diagram of a conventional DLL is shown in Figure 2.1. There are three major parts: (1) phase detector (PD) (2) a variable delay line (3) controller. The PD compares phase error between reference clock and output clock; meanwhile, it sends output signal to adjust delay time of the delay line. When DLLs achieve locked, ideally, there is no phase error between reference clock and output clock. Equation (2.1) shows the relationship of reference clock and output clock, where K is an integer, T_{ref} represents the clock period of the reference clock. T_d denotes the delay time of delay line.

$$K \times T_{ref} = T_d \qquad (2.1)$$

The design challenge of the DLL is to consider process, voltage, temperature (PVT) variations, clock jitter, power consumption, portability, lock time and so on. Therefore, different types of DLLs have been proposed to reach some of these objectives. We will briefly introduce and classify these delay-locked loops.

Analog DLLs have advantages of the smaller static phase, good jitter performance, and fine resolution because the delay is varied continuously. However, it suffers from slow locking time and sensitivity to PVT variations, which is not suitable for System on Chip (SoC) environment. On the contrary, the digital DLLs are more robust against PVT variations, providing fast lock time and easy to design. However, the quantization error of the digital DLLs is unavoidable because the delay adjustment is in a discrete manner. Recently, mixed-mode DLL have been proposed in [1]. They can possess both the advantages of analog DLLs and digital DLLs, such as better jitter performance of analog part and fast-lock operations of digital search algorithm. However, their power consumption is still larger than all-digital DLL, and it is hard to integrate digital and analog blocks simultaneously. We simply classify DLLs in three different types, they are：

(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-mode DLL: Use 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.

2.2 DESIGN OF ANALOG DELAY-LOCKED LOOP

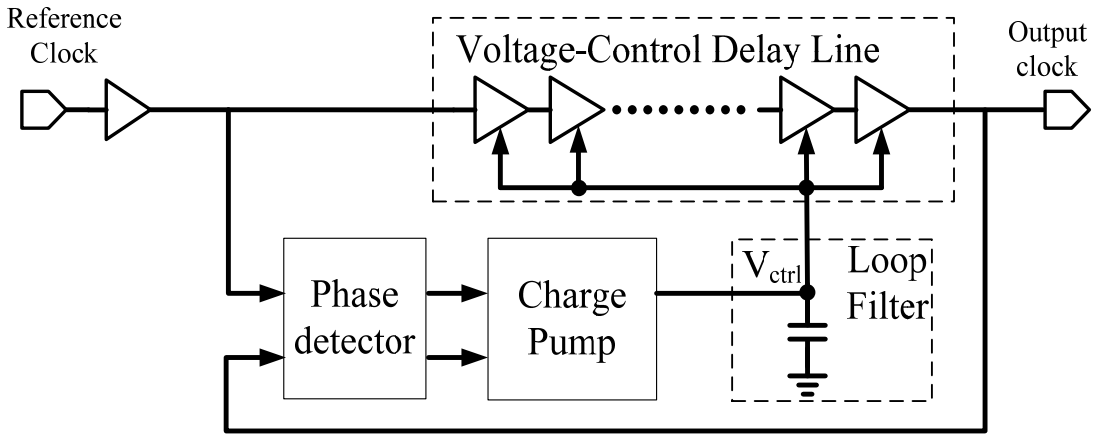


Figure 2.2 Block diagram of analog DLL

Figure 2.2 illustrates the block diagram of an analog DLL. It consists of 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. The phase detector compares the phase error between the reference clock and output clock, which is the delay version of VCDL, and produces an up/down signal to the charge pump. The charge pump integrates the phase error, and the loop filter produces a control voltage, V_{ctrl}, to decrease or increase the delay time of the VCDL. Once the reference clock is aligned with the output clock, the DLL is in the lock state.

2.2.1 Stability Analysis of Delay-Locked Loop [2]

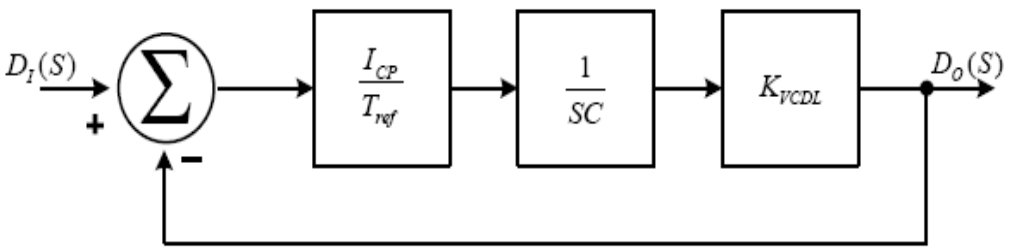


Figure 2.3 Small signal AC model of the analog DLL.

Before starting the stability analysis of ADLL, the small signal AC model shall be introduced first. Figure 2.3 shows this model, the summer stands for phase detector, Icp 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

$$\frac{D_0(s)}{D_1(s)} = \frac{1}{1+\frac{s}{\omega_N}} \tag{2.2}$$

Where

$$\omega_N = \frac{I_{CP} \times K_{VCDL}}{T_{REF} \times C} \tag{2.3}$$

From Eq. 2-2, 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 ωN, is much smaller than the phase detector comparison frequency (generally 10:1). Therefore, the following equation should be satisfied for stability consideration.

$$\frac{\omega_N}{\omega_{REF}} = \frac{I_{CP} \cdot K_{VCDL}}{2\pi \cdot C} \leq \frac{1}{10} \tag{2.4}$$

Where

$$\omega_N = \frac{2\pi}{T_{REF}} \tag{2.5}$$

2.2.2 Jitter Analysis of Delay-Locked Loop [3]

Due to the noise of the real world, the output clock edge may have some timing uncertainty or fluctuation, which is called clock jitter. There are many factors will influence clock jitter performance, such as thermal, supply and substrate noise. We describe three major definitions of the jitter below. Suppose Tn is the nth clock period, $\overline{T}$ is the mean value of the clock period. Also, Figure 2.4 illustrates the long-term jitter and the cycle-to-cycle jitter.

(1) The absolute jitter or long-term jitter: $\Delta T_n = T_n - \overline{T}$,the quantity ΔTn is an indication of jitter.

$$\Delta T_{abs}(N) = \sum_{n=1}^{N} \Delta T_n \tag{2.6}$$

(2) The Cycle jitter: The rms value of the timing error ΔTn

$$\Delta T_c = \lim_{N\to\infty} \sqrt{\frac{1}{N}\sum_{n=1}^{N} \Delta T_n^2} \tag{2.7}$$

(3) The cycle-to-cycle jitter: The rms difference between two consecutive periods

$$\Delta T_{c-c} = \lim_{N\to\infty} \sqrt{\frac{1}{N}\sum_{n=1}^{N} \left(\Delta T_{n+1} - \Delta T_n\right)^2} \tag{2.8}$$

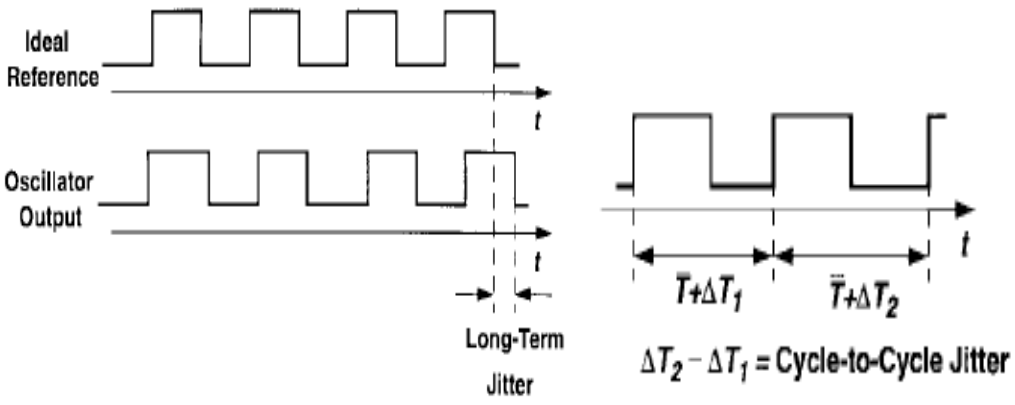


Figure 2.4 long-term jitter and cycle-to-cycle jitter

2.2.3 Voltage-controlled Delay Line

Delay elements are essential parts for clocking operation in high speed VLSI application. The delay of each delay element is proportional to its RC time constant. By changing the effective resistance or capacitance, delay elements can adjust their delay time. However, the characteristics of the voltage-controlled delay element are sensitive to supply noise and PVT variations. Here, we will introduce the three common approaches of VCDL. They are current-starved delay line (CSDL), RC-time-constant delay line (RCDL), and differential voltage-controlled delay line (DVCDL).

1. Current-Starved Controlled Delay Line

A basic delay element of CSCDL is shown in Figure 2.5. 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.

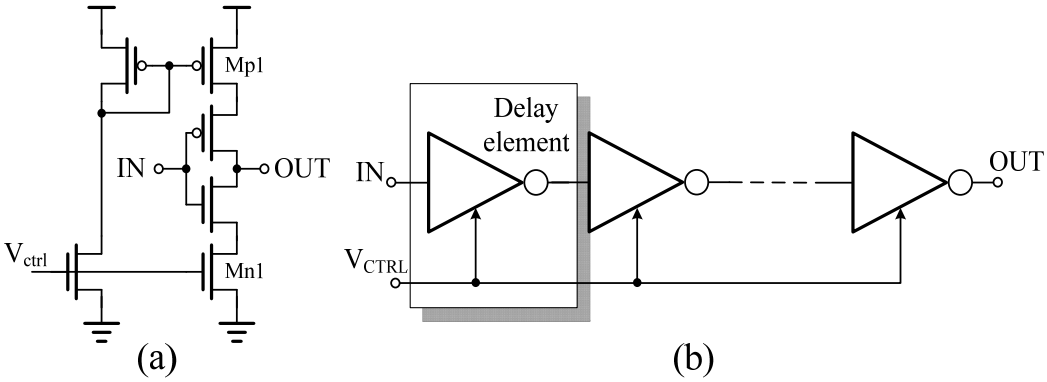


Figure 2.5 The CSDL (a) delay element (b) delay line

2. RC-time-constant Delay Line

The basic delay line of RC-time-constant controlled delay element is shown in Figure 2.6(b). The circuit can be obtained by cascading even number of the same delay

elements. In Figure 2.6 (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.

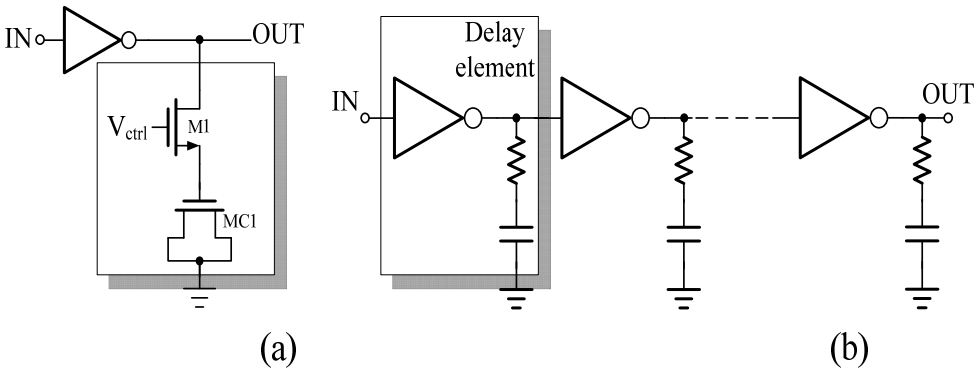


Figure 2.6 The RCDL (a) delay element (b) delay line

3. Differential voltage-controlled Delay Line

In order to achieve better common-mode noise rejection, the differential delay element has been introduced in [4] as shown in Figure 2.7. The delay element consists of a pseudo differential amplifier for high-speed operation. Since the strong PMOS latch will decrease the bandwidth of the delay cell, the weak PMOS latch, M3–M4, is adopted to speed up the signal transition of the differential pair, M1–M2 .However, it will slow down the slew rate of the output signal. To improve the driving capability of the delay cell, the PMOS transistors, M5–M6 , in triode are added to increase the driving capability. The PMOS transistors, M7–M8, are used to adjust the delay.

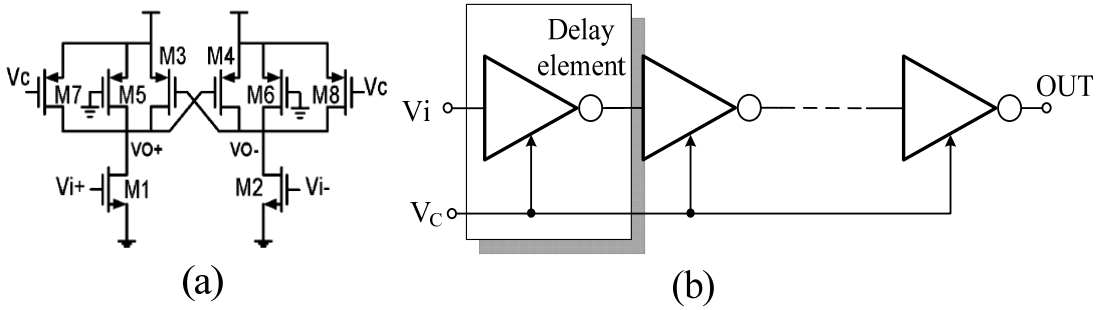


Figure 2.7 The DVCDL (a) delay element (b) delay line

2.2.4 Phase Detector

Phase Detector is a circuit that is response the relationship between reference and feedback signal. Figure 2.8 shows three-state phase detector circuit and Figure 2.9 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.

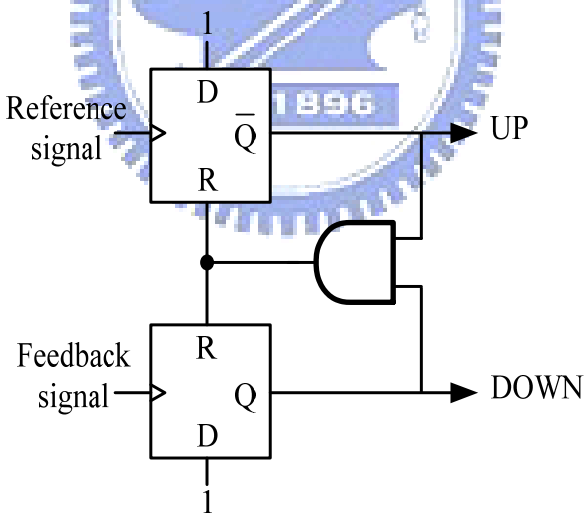


Figure 2.8: Three-state phase detector

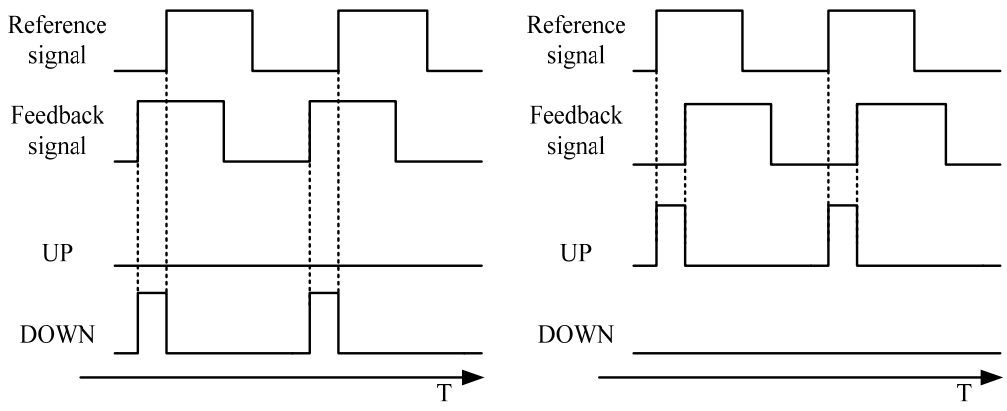


Figure 2.9: PD responses with (a) reference signal lag feedback signal (b) reference signal lead feedback

The delay time to reset all internal nodes limits the maximum operation frequency of the phase detector. Therefore, a dynamic phase detector [] is proposed to solve this problem as shown in Figure 2.10. The phase detector is composed of two half-transparent (HT) registers. Due to symmetry of circuit and only three gate delay of critical path, this phase detector has less phase offset and can be operated at a higher frequency than the conventional phase detector.

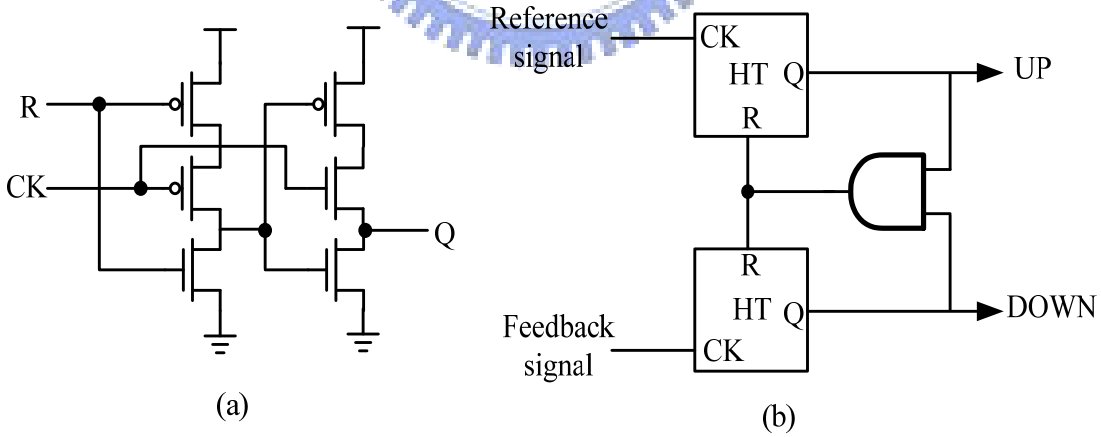


Figure 2.10: (a) The Half-transparent register (b) dynamic PD

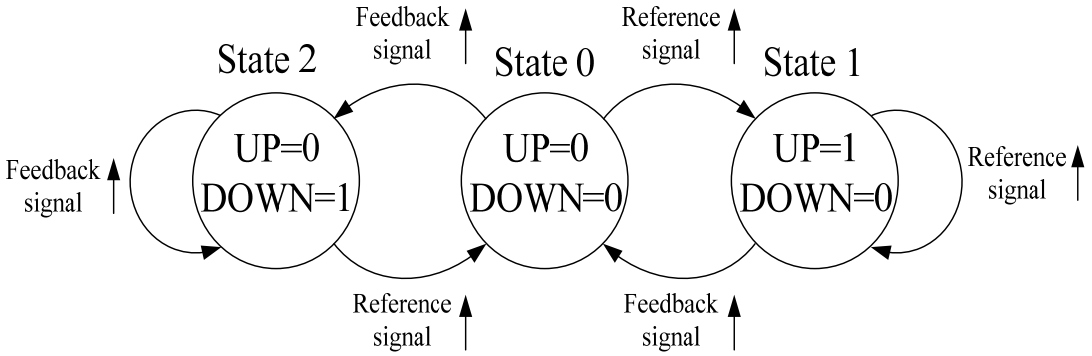


Figure 2.11: PD state diagram

In the Figure 2.11, 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. +2pi,-2pi. A phase difference larger than 2pi is truncated with respect to integer of 2pi. 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.

2.2.5 Charge Pump and Loop Filter

The simple model of charge pump and loop filter is shown in Figure 2.12. 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. Figure 2.13 shows the simplified timing diagram of charge pump.

The loop filter can be either passive or active. In general, a passive filter is simple to design and has better noise performance. The passive filter was shown in Figure 2.14, 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.

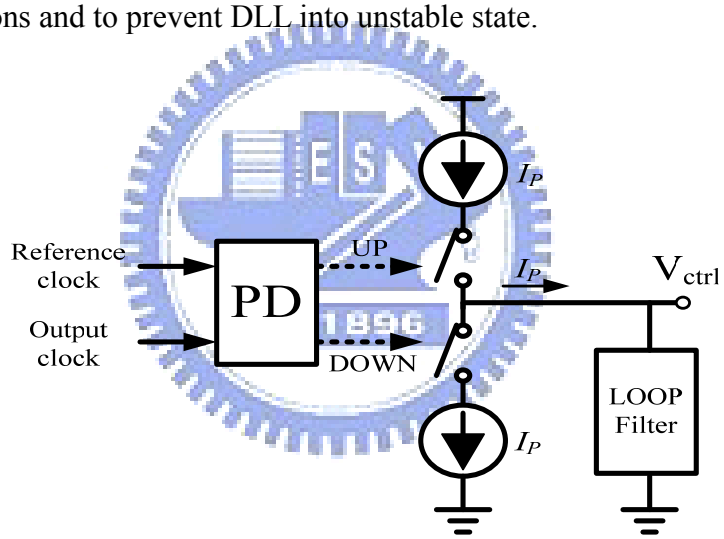


Figure 2.12: Simple model of charge pump and loop filter

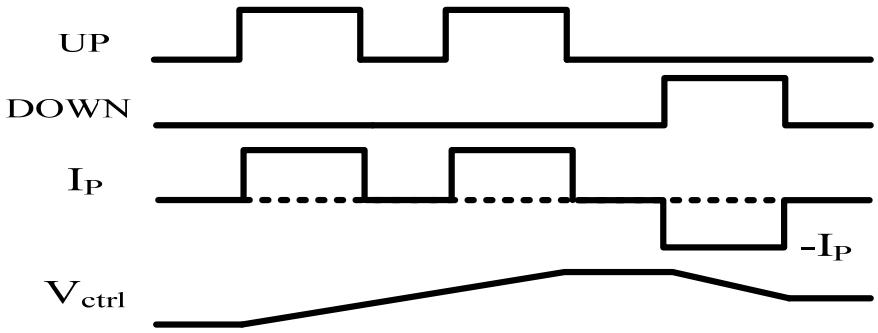


Figure 2.13: The simplified timing diagram of charge pump

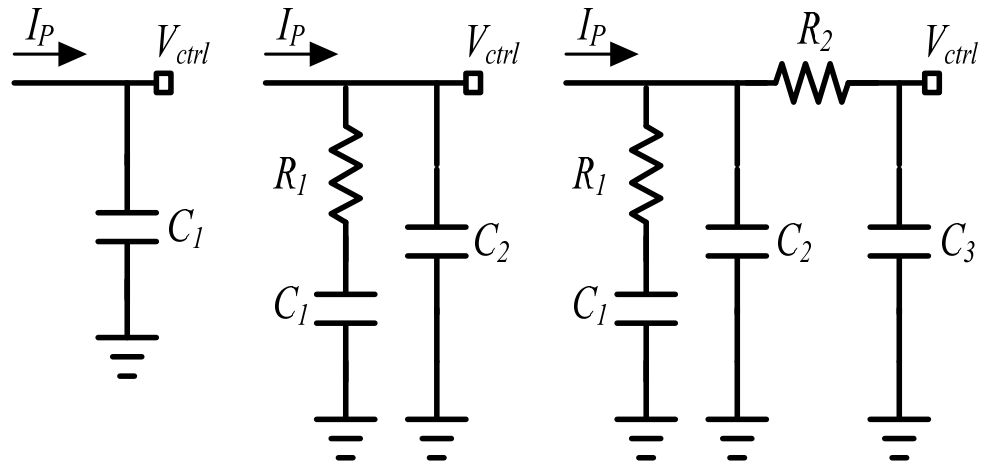


Figure 2.14 Loop filter

2.3 DIGITAL DELAY-LOCKED LOOP OVERVIEW

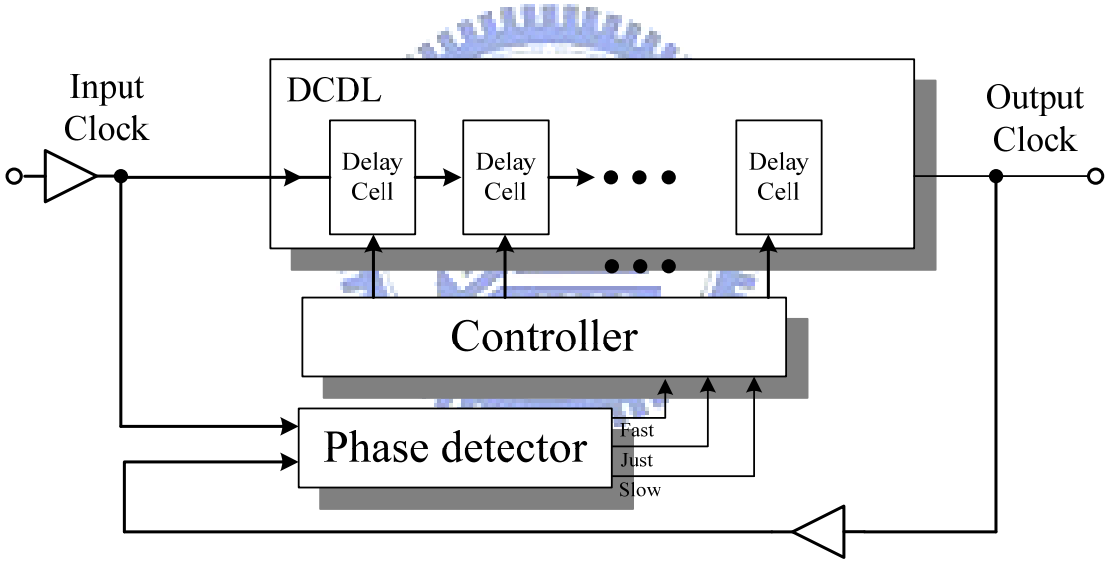


Figure 2.15 Block diagram of digital DLL

As the era of System-On-Chip (SoC) coming, people give more attention on digital DLLs since they are easy to port to different process. Recently, due to the fuel crisis, a power-saving issue becomes more and more important. The power of CMOS circuit is determined primarily by equation $P=CFV^2$, which decreases quadratic ally with supply voltage. For this reason, the digital DLLs have better power-saving performance since it can operate at lower voltages than analog DLLs.

The conventional digital DLL block is shown in figure 2.15. It consists of three major blocks. They are phase detector (PD), control unit (CU) and digital control delay line (DCDL). The PD detects the phase error between the input clock and the output clock and generates signal to the CU. The CU adjusts the delay line making input clock and output clock in-phase.

We classify four major types of digital DLLs; they are register-controlled, counter-controlled, successive approximation register-controlled, and time measurement controlled DLLs. The following section will describe in detail.

2.3.1 Register-controlled DLL [5]

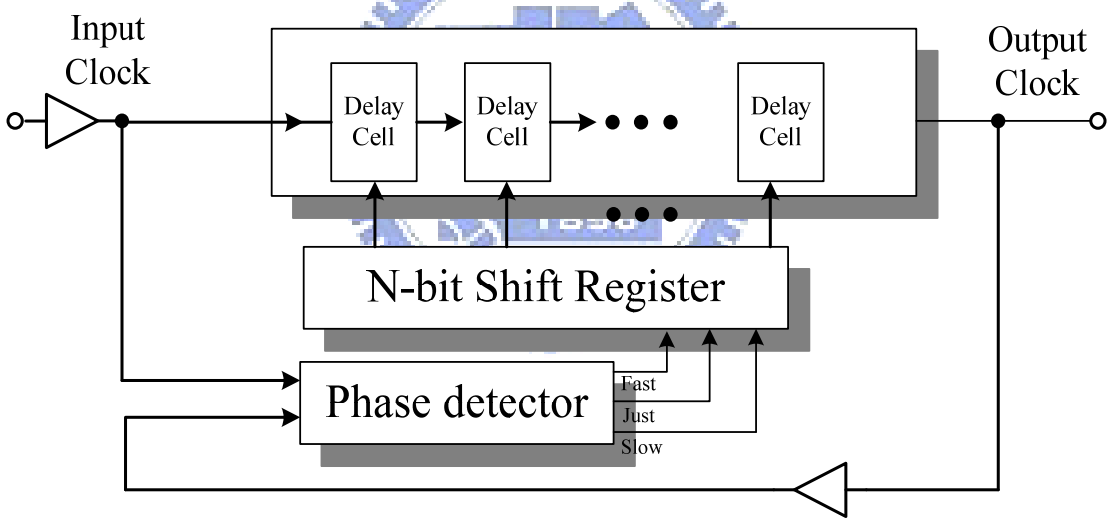


Figure 2.16 Register controlled DLL

Figure 2.16 shows the block diagram of register-controlled DLL. The n-bit shift register controlled by the output of phase detector. At any time, only on 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. The detail operation is described as following. 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. Thus, it increases the chip area and power consumption. Beside, the control mechanism is one by one, which 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.

2.3.2 Counter-controlled DLL [6]

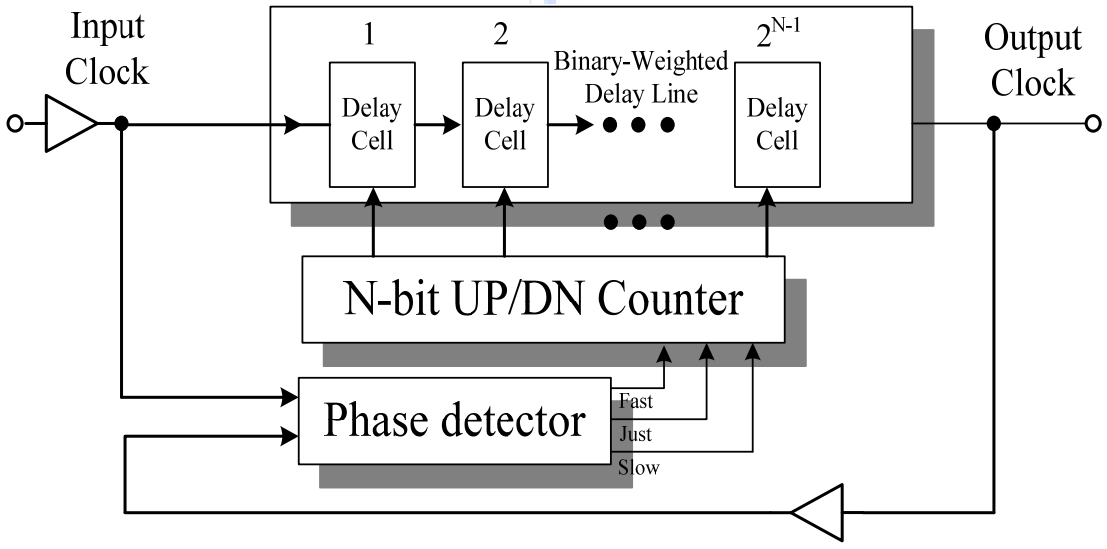


Figure 2.17 counter-controlled DLL

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. Figure 2.17 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.

2.3.3 Successive Approximation Register-controlled DLL [7]

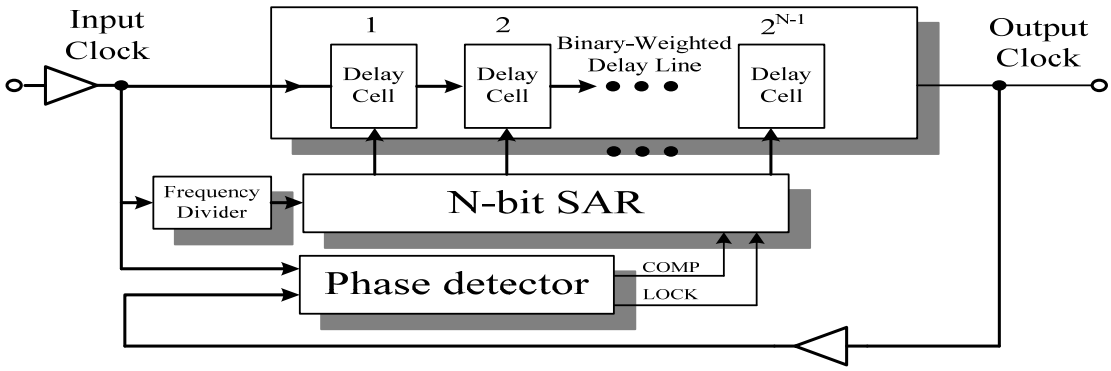


Figure 2.18 SARDLL

The locking time is an important parameter for digital DLL to evaluate the performance, especially in the high-speed memory applications. The DLLs that mentioned above based on the linear search exhibit the same lock time. The linear search algorithm; however, increases the locking time when frequency is wider. The binary search algorithm may be a better method to reduce the locking time. Figure 2.18 shows the block diagram of binary search (SAR) DLL.

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 2.19 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".

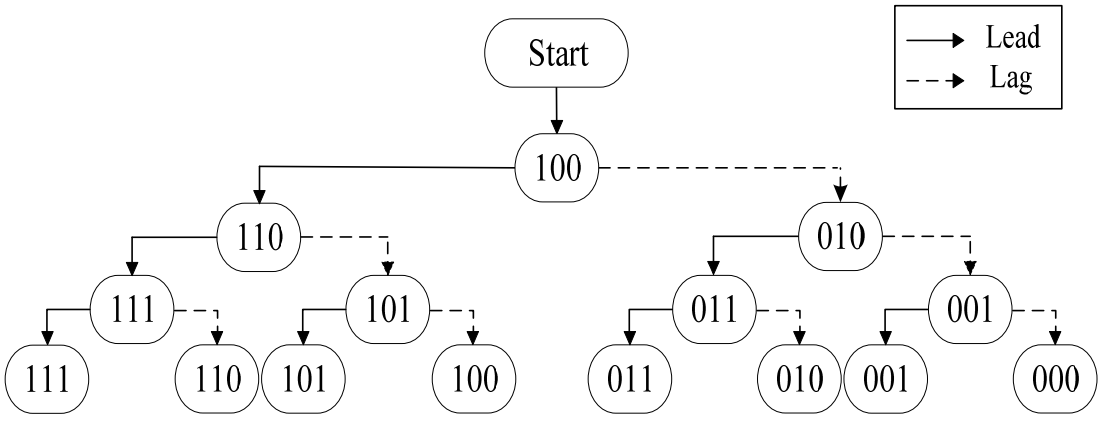


Figure 2.19 Flowchart of 3-bit binary search algorithm

The SAR DLL 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 log2(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 [8] 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.

2.3.4 Time measurement controlled DLL

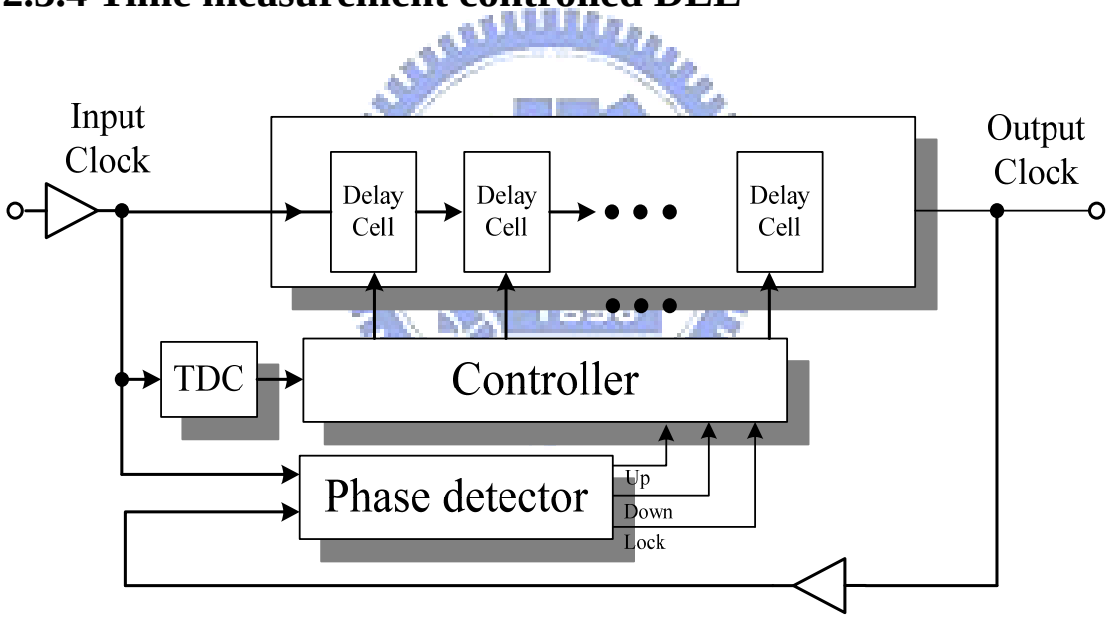


Figure 2.20 TMDLL

Another mechanism to reduce the locking time was proposed in [9] as shown in Figure 2.20. The time measurement controlled 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 as shown in Figure 2.21. The TDC is used in ADPLL or ADDLL to convert timing information directly to the digital code. The TDC usually consist of the delay that is identical or multiple or fractional to the single delay

cell in the delay line or oscillator, the concept is let timing signal to pass this delay then extracting the information to the digital code.

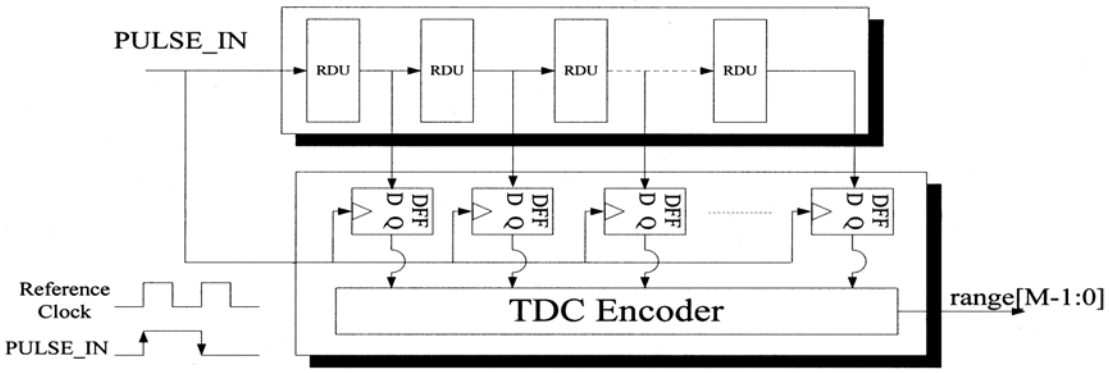


Figure 2.21TDC circuit

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 is still the area requirement.

2.4 CLASSIFICATIONS OF 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). The most common delay time adjustment is usually realized by a path-selection approach as shown in Figure 2.22 [10]. In this example, 2^n delay buffer are connected in series. A decoder decodes an n-bit control word D into 2^n control lines. Hence, if the propagation delay time of each buffer stage is T_{buffer}, then the time resolution is $2*T_{buffer}$.

There are other architectures that have been used to implement a DCDE. In the following section, we will introduce different kinds of DCDE.

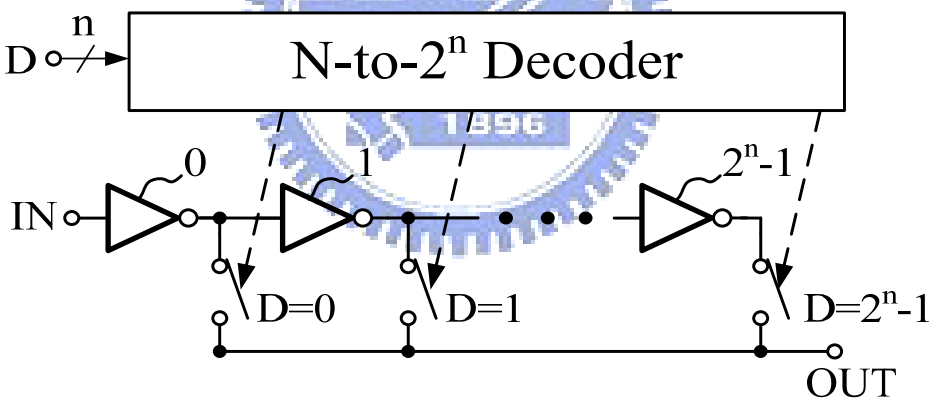


Figure 2.22 DCDL realized by a path-selection method

2.4.1 Shunt Capacitor based DCDE

Figure 2.23 shows the basic circuit of using a shunt capacitor based DCDE [11]. In this circuit, MC1~MCn acts as shunt capacitor. Transistor M1~Mn controls the charging and discharging current to the MC1~MCn. The operating is similar to RCCDL; replace the Vctrl 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-weigh. The drawback of shunt capacitor based DCDE is sensitive to power supply noise and PVT variation.

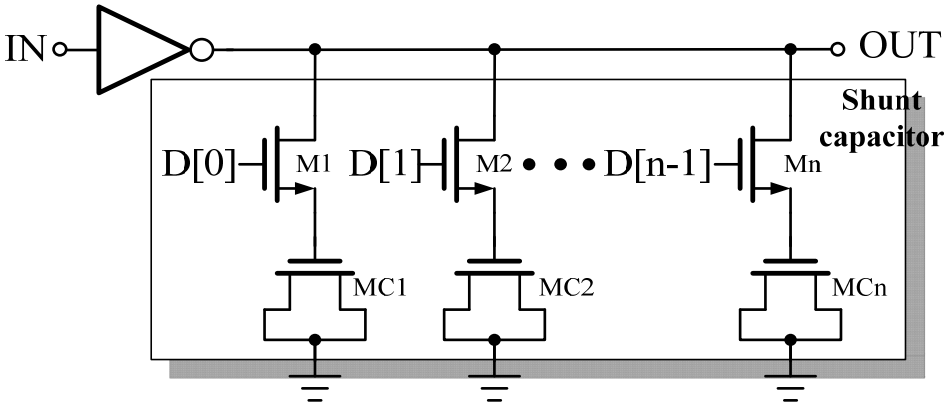


Figure 2.23 Shunt-capacitor based DCDE

2.4.2 Standard Cell based DCDE

One simple example of standard cell based DCDE was proposed in [12], as shown in Figure 2.24. 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.

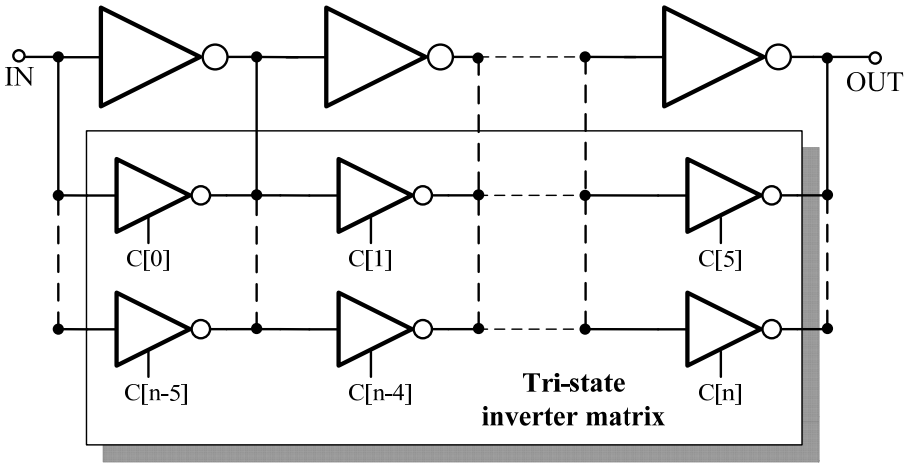


Figure 2.24 Parallel tri-state inverter based DCDE

The other example, as shown in Figure 2.25, 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 [13]. 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 [12]. 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.

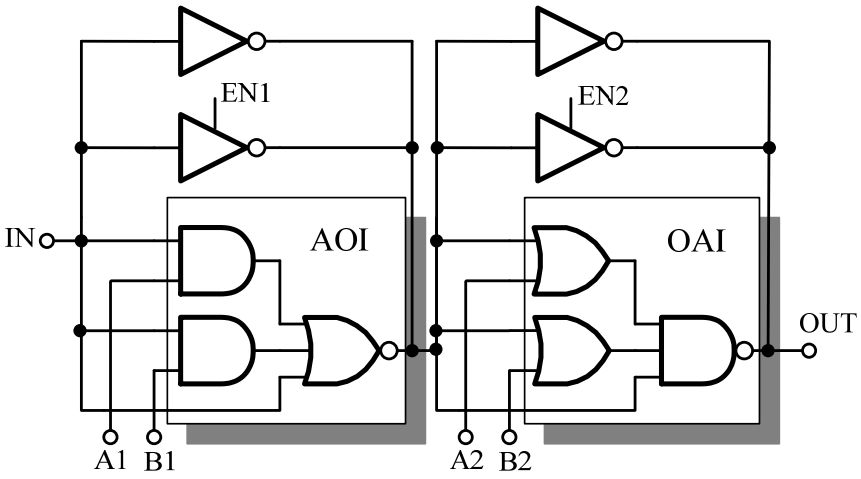


Figure 2.25 AOI-OAI parallel based DCDE

Due to inherent drawbacks of cell-based delay cell, such as finite resolution and nonlinear delay versus control words, the novel delay cell is proposed in [13] to improve delay resolution and monotonic delay behavior with respect to digital control codes. Figure 2.26 (a) illustrates a novel delay cell using a two-input NOR gate. The delay method is to give node'D different digital signal since total gate capacitance of transistors M2 and M3 varies with different voltage of node'D. Figure 2.27 compares the delay resolution between [12] and [13], showing that the novel delay cell [13] has better linearity than OAI cell in [12].

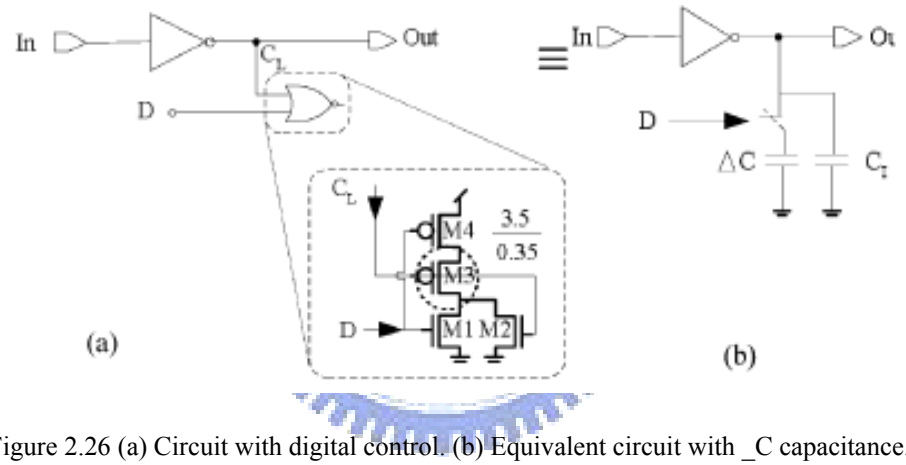


Figure 2.26 (a) Circuit with digital control. (b) Equivalent circuit with _C capacitance.

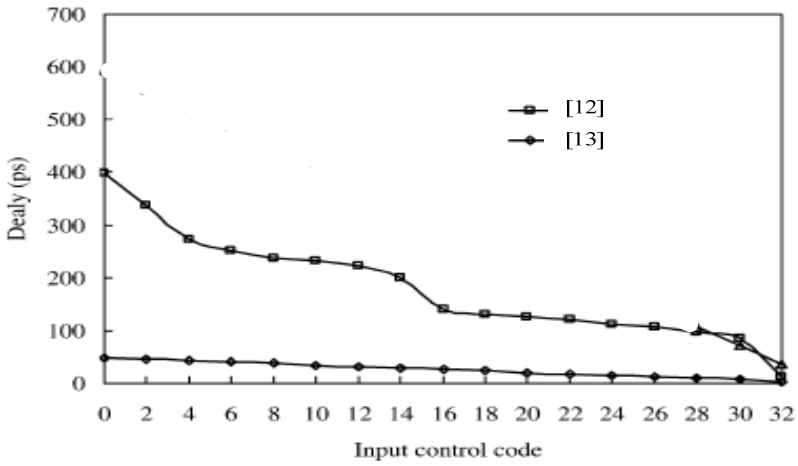


Figure 2.27 Comparisons among the proposed DCV and other approaches

2.4.3 Low power DCDE

The low power DCDL named binary-weighted differential-delay cells (BWDC) is proposed in [14], as shown in Figure 2.28. 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. The difference of capacitance determines the finest delay resolution, which can be made sufficiently small. The BWDC also has two distinct features 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.

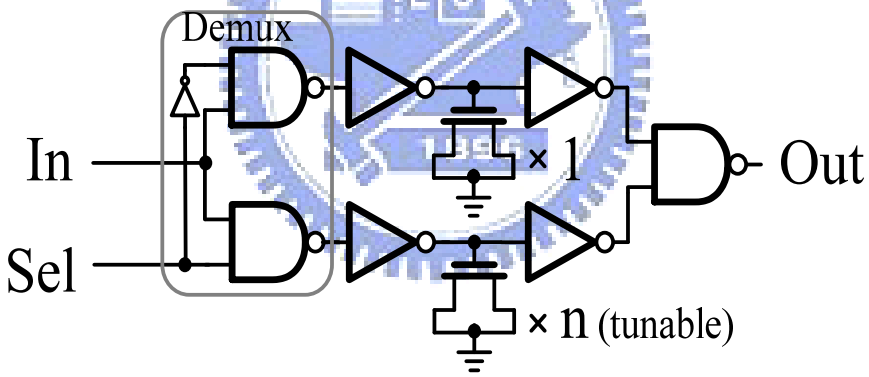


Figure 2.28: The binary-weighted differential-delay cell.

2.4.4 Current-starved based DCDE

The current starved based DCDE was proposed in [15]. As Figure 2.29 shows, the charging and discharging currents of the inverter, composed of M1 and M2, are controlled by two sets of current-controlling nMOS (Mn0, Mn1, …) and pMOS (Mp1, Mp2, …) transistors at the source of M1 and M2, respectively. The current controlling transistors are sized in a binary version to allow binary-weighted delay time. 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.

However, one of the problems with the current staved based DCDE architectures is the non-monotonic delay behavior with ascending binary input vector. 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.

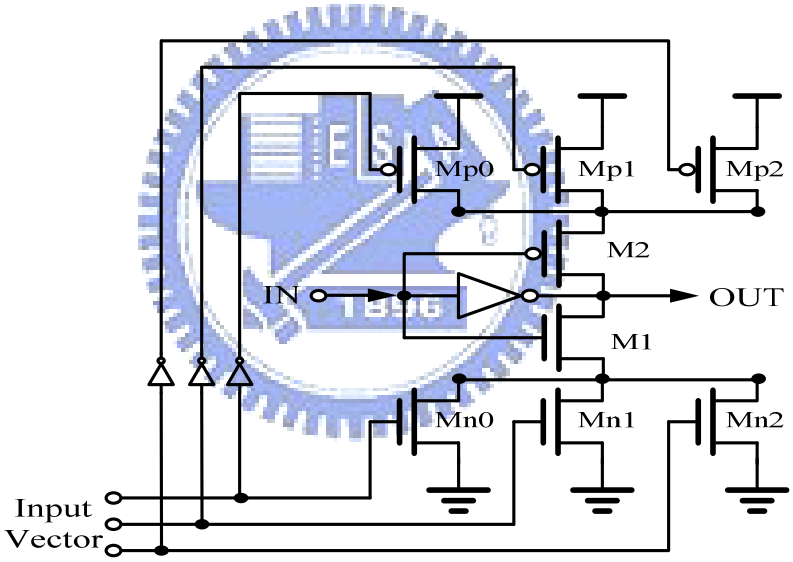


Figure 2.29 Current starved based DCDE

In [16], there are two factors depending on the input vector to affect the delay :

(1) The resistance of the controlling transistors:

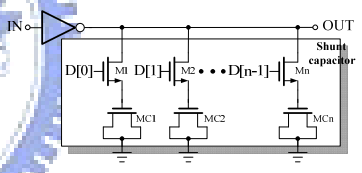
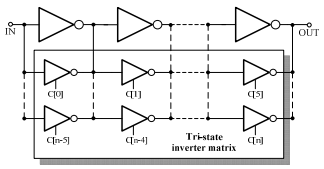
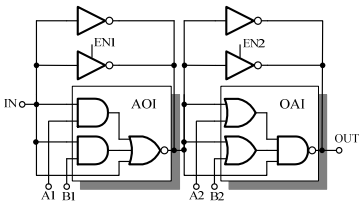
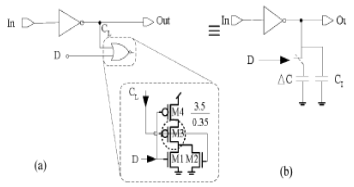
The circuit delay can be increased / decreased by increasing / decreasing the effective ON resistance of the controlling transistors at the source of M1.

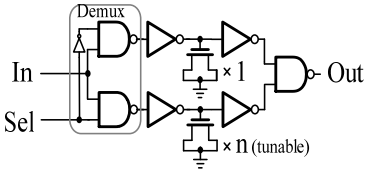
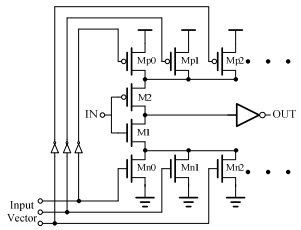
(2) The capacitance of the controlling transistors:

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	➢ Sensitive to power supply noise ➢ Process mismatch	
Standard-cell based	➢ Larger area and power dissipation ➢ Delay resolution ➢ Different coarse delay and fine delay cells	  

Low power DCDL	➢ Intrinsic delay time as delay cell cascaded	
Current-starved based	➢ Sensitive to PVT variation ➢ Poor linearity	

2.5 COMPARISON OF DIFFERENT TYPES OF DLLS

Finally, comparison of different types of DLLs is given in Table 2

Table 2 Comparison of different types DLL

	Analog DLL	***Mixed-mode DLL***	***Digital DLL***
Design cycle	Slow	Slow	Fast
Noise rejection	Poor	Poor	Good
Resolution	High	High	Low
Lock time	Slow	Middle	Fast
Area	Large	Large	Small
Power consumption	Large	Large	Small

CHAPTER 3
MULTIPHASE DLL AND DLL-BASED FREQUENCY MULTIPLIER

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 they have ability to generate multiphase clock signals. The multiphase clock signals can be used in various applications such as time-interleaved architectures, transmitter and receiver [9], and high speed IO. 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 [24]. In frequency synthesizer, the multiphase DLL can combine each phase and generate a high frequency signal [20].

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 due to their ease of design, better immunity to on-chip noise, and no jitter accumulation characteristic.

3.2.1 DDR SDRAM Controller Application [17]

In Double Data Rate (DDR) SDRAM controller design, output data strobe (DQS) signal must be delayed by a fixed timing delay (tSD) to capture the output data (DQ) correctly. Figure 3.1 shows this read operation timing budget. Ideally, the DQS and

DQ is edge aligned by DDR SDRAM. However due to pin-to-pin skew among all DQ and DQS, and PCB board skew, the data valid window becomes smaller than expected. The calculations for timing budget show that the optimal value for tSD is approximately 20 percent of an input clock period. Since the input clock frequency range from 100MHz to 200MHz (DDR-200/266/333/400), the tSD value varies from 2ns (=10nsX0.2) to 1ns (=5nsX0.2). Therefore, a five-phase all-digital DLL is proposed to generate the desired tSD delay for DQS signal.

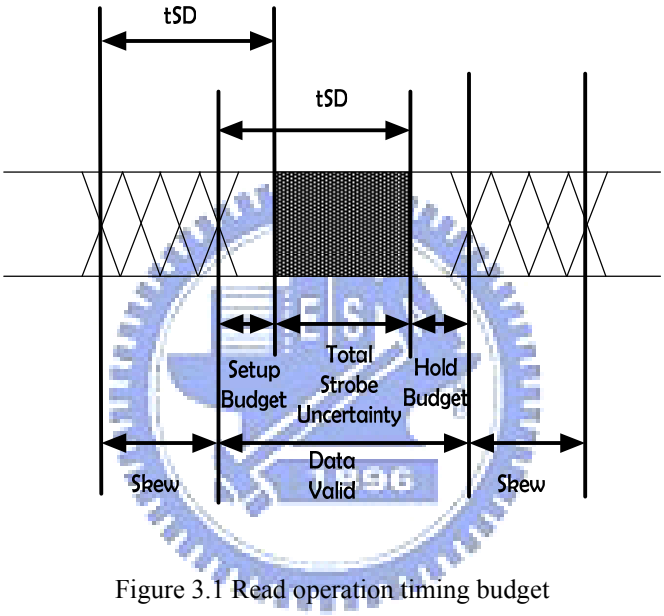


Figure 3.1 Read operation timing budget

The block diagram of the five-phase all-digital DLL for a DDR SDRAM controller application is shown in Figure 3.2. 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 $2*T_{FREF}/5$, where T_{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 $1.2xT_{FREF}$, which means the phase shift between DQS and DQSD is still $0.2xT_{FREF}$. As a result, the desired tSD delay can be generated by the multiphase DLL.

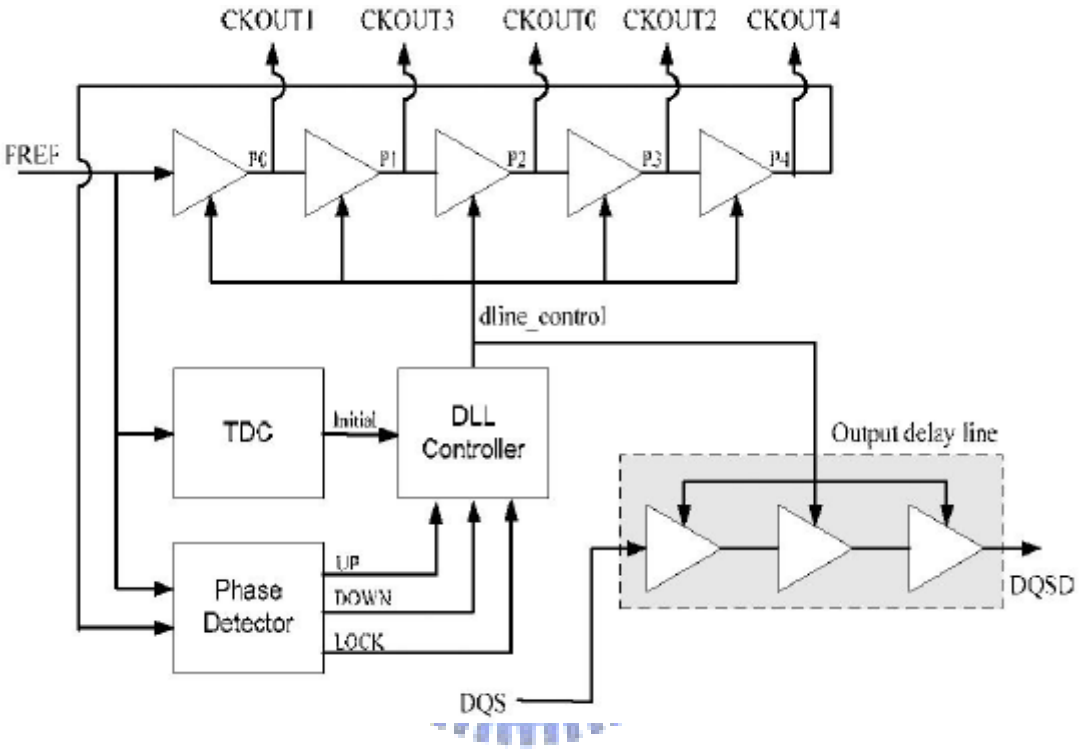


Figure 3.2 Architecture of multiphase DLL for DDR SDRAM application

3.2.2 Transmitter [9]

In digital communication applications, the multiphase DLL is applied to a data cannel compression transceiver. The architecture of the transceiver is shown in Figure 3.3. 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 "TX delay mirror" shown in Figure 3.3(a) is used to compensate the delay time of the parallel-to-serial converter. The receiver shown in Figure 3.3 (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 3.3 (b) is used to estimate the accurate delay of TREF/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.

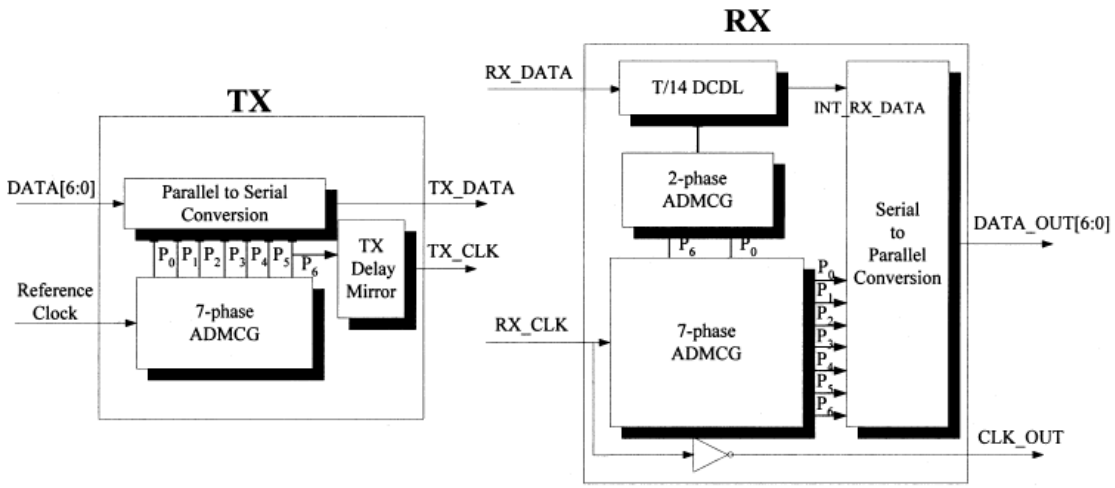


Figure 3.3 7:1 Data channel compression transceiver. (a) Transmitter circuit. (b) Receiver circuit.

3.2.3 High-performance Microprocessors [18]

Execution engines of multi-gigahertz superscalar processors require multiphase clock signals with accurate edge-position to trigger dynamic data path circuits. The intermediate clock phases need automatic stretch ability in proportion to the core clock period. Moreover, multiphase clock can also be used in large SoCs to achieve a higher operating speed than the main internal clock frequency. Figure 3.4 shows the organization of the multiphase clock generator for high-performance microprocessors. This multiphase clock generator has several features: adopts a closed-to-open loop control scheme with a jitter-averaging counter to combine the benefits of both fixed/programmable delay-lines and digital DLL-based techniques, thereby achieving high resolution, flexibility, and frequency-range.

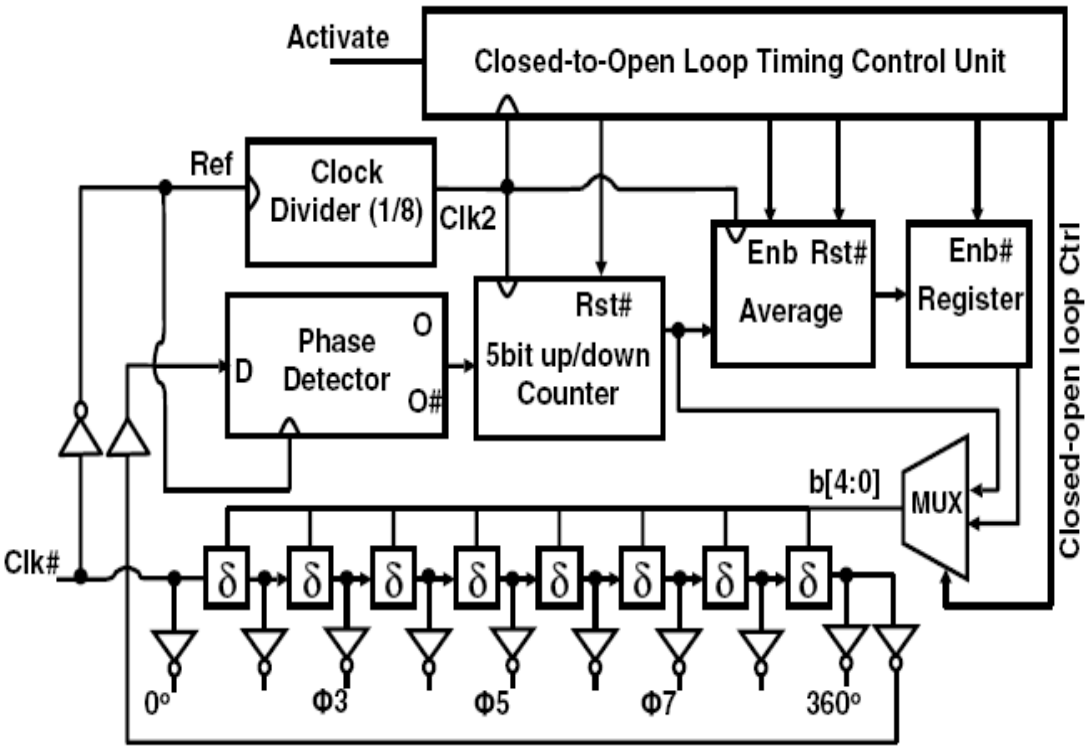


Figure 3.4 Organization of the multiphase clock generator.

3.3 INTRODUCTION OF DLL-BASED FREQUENCY MULTIPLIE

PLL-based frequency synthesizer has been widely used until recent times. Another type that draws attention is DLL-based frequency synthesis. DLL-based frequency synthesizer takes advantage of low phase noise since they derive the output signal directly from a clean crystal reference which has low noise accumulation. Moreover, the DLLs can be designed as a first-order system to allow a wider loop bandwidth and settling times on the order of nanoseconds, which are particularly suitable for applications where fast-lock issue is required, such as ultra-wide band system.

3.3.1 Basic of the DLL-Based Frequency Multiplier

Conventional DLLs can not generate different frequency of input clock frequency. However, by using the edge combiner and replacing voltage-controlled oscillator (VCO) with delay line, a DLL can operate as PLL. The simplified block diagram of DLL-based frequency synthesizer is shown in Figure 3.5. When the loop is locked, the output phases of every delay stage are evenly spaced one reference clock period Tref. Each phase difference of two delay stage has a delay of Tref/N. Then, the edge combiner can generates a transition for each phase output transition; hence, the output frequency is N times the reference frequency Tref. The operation diagram is illustrated in Figure 3.6.

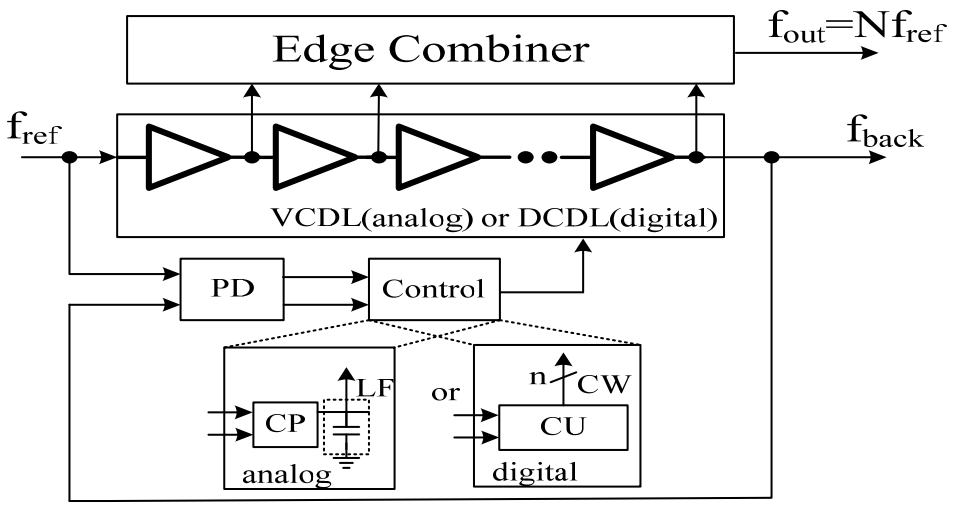


Figure 3.5 Organization of the multiphase clock generator.

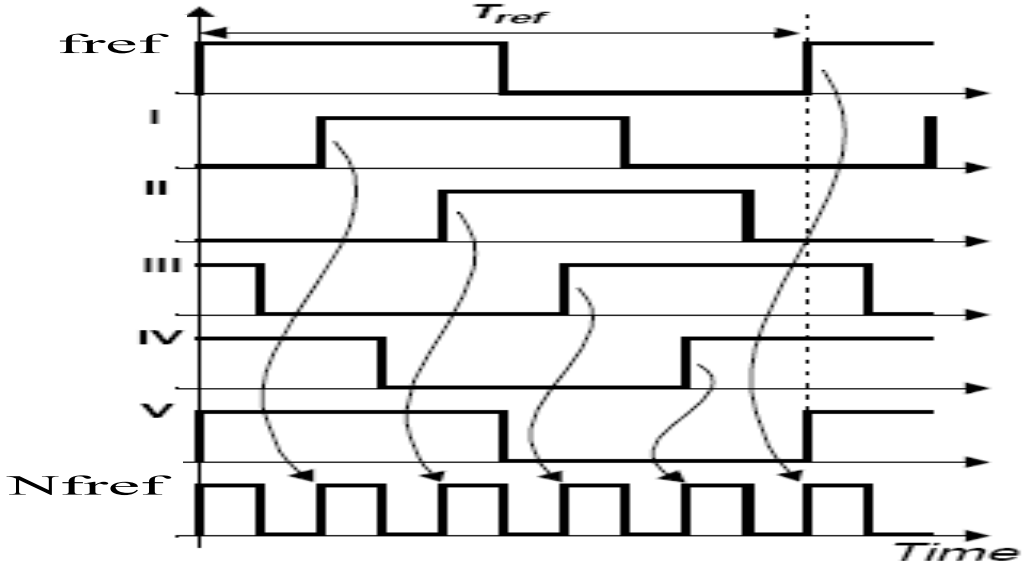


Figure 3.6: Operation timing diagram of frequency multiplier

3.3.2 Timing Jitter Accumulation [19]

A DLL-based frequency using a voltage-controlled delay chain has an inherent advantage over a PLL using a voltage-controlled oscillator. Figure 3.7 shows timing jitter accumulation for an oscillator compared with that of a DLL-based frequency multiplier. In an oscillator, random timing errors accumulate because the timing jitter at

the end of each oscillation is the starting point of the next. The random timing error of the output signal is the sum of the timing errors of all previous oscillations. This translates to a poor long-term jitter performance, or, equivalently, poor close in phase noise. Figure 3.7 shows the random timing uncertainty for a ring oscillator increase as a function of time.

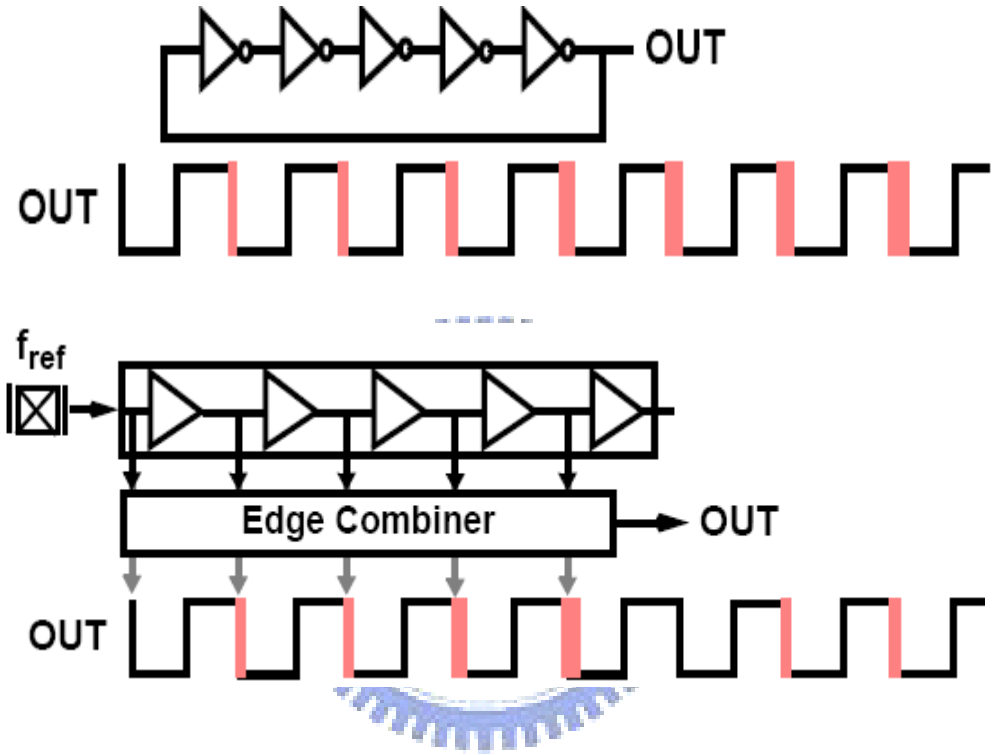


Figure 3.7 Timing jitter accumulation for ring oscillator and delay chain.

In contrast, for a finite length delay line in the DLL-based frequency multiplier, the random timing error accumulates only within a single delay chain cycle. The timing error in one cycle of the delay chain does not affect the next cycle, because the waveform that triggers the next output oscillation is the reference clock waveform. This provides excellent long-term jitter performance, or, equivalently, a low close-in phase noise. The different phase noise signatures of a PLL with a VCO and a DLL-based frequency multiplier can also be understood by examining the source of synthesized output waveform. In a PLL, the output signal is taken directly from a VCO whose timing uncertainties accumulate over many oscillation cycles, limited by the time

response of the PLL in which it is embedded. However, the PLL bandwidth is constrained by practical considerations to a value several orders of magnitude lower than the output frequency. In contrast, each output edge from the DLL only contains the timing uncertainties accumulated from the previous delay stages within the same reference oscillation period. Limited jitter accumulation gives a flat phase noise profile for offset frequencies less than f_{ref}. The long-term timing error accumulation, equivalent to the close-in phase noise, is much lower than that of a typical VCO.

3.4 APPLICATION OF DLL-BASED FREQUENCY MULTIPLIE

Recently, DLL-based frequency multipliers are used in communication system applications, high speed IO due to its low phase noise. In this section, we will introduce several DLL-based frequency multiplier architectures and its applications.

3.4.1 Local Oscillator for PCS Application [20]

Figure 3.8 shows the DLL-based frequency multiplier for PCS application. The objective of the DLL-based frequency multiplier is to produce a low-phase-noise RF signal by taking advantage of the inherently low jitter of a low-frequency crystal oscillator reference. As shown in Figure 3.8, the reference crystal signal is first amplified and drives the delay chain. The delay chain consists of nine delay stages and is locked to half of the reference crystal period. A phase detector, charge pump, and loop filter are used to create the control voltage for the delay chain. The edge combiner combines all the outputs from the DLL and creates the multiplied frequency.

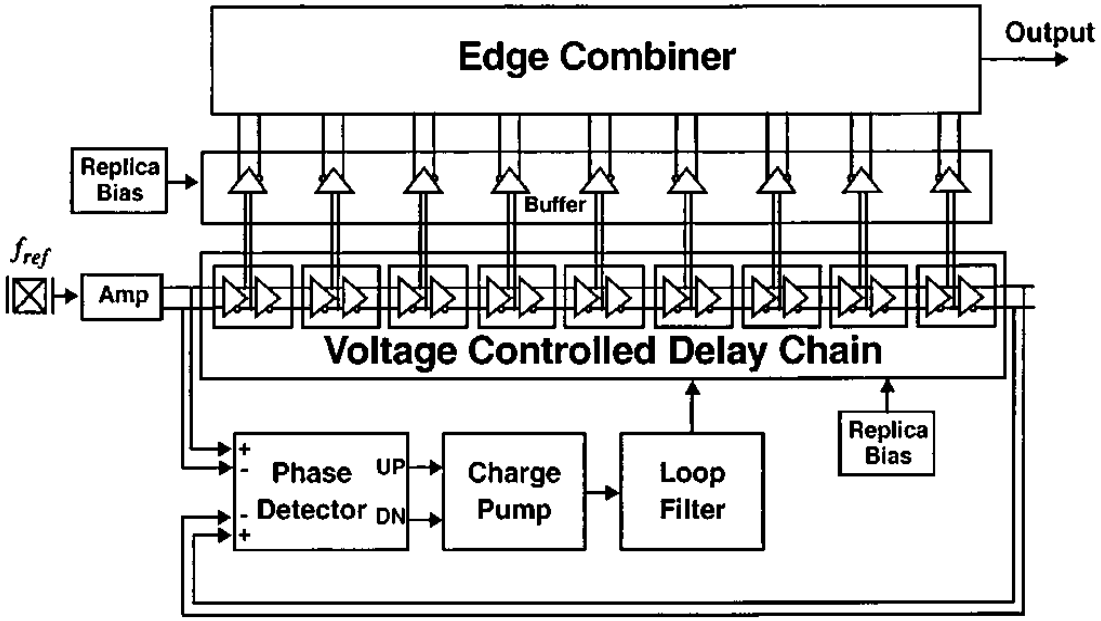


Figure 3.8 Block diagram for the experimental prototype

The edge combiner consists nine NMOS input differential pairs and a pair of LC-tanks and is similar to the folding amplifier as shown in Figure 3.9. The DLL have an odd number of delay element, and a single current is modulated back and forth between the LC-tanks to create the multiplied frequency. However, the multiple factor is fixed, and LC-tanks occupy a lot of chip area, which is not suitable for SoC design.

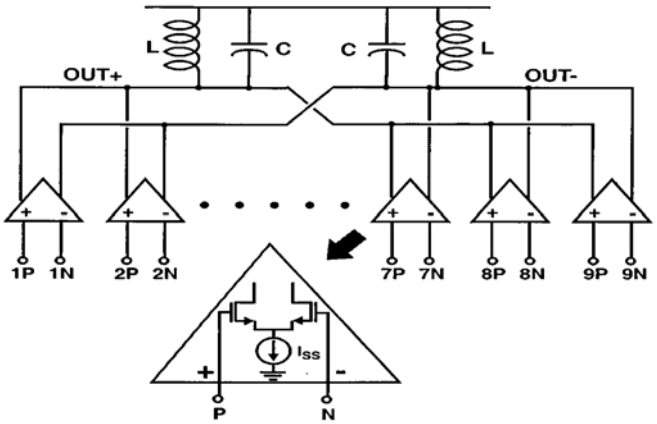


Figure 3.9 Edge combiner schematic

3.4.2 A Multiplying DLL for High Speed On-chip Clock Generation [21]

As bandwidth demand of computer and digital communications components continues to grow, high-speed serial I/O links are replacing traditional parallel buses. Operating at speeds of up to 5 Gb/s, such high-speed I/O circuits are already found in packet switches, circuit switches, and processor- memory interconnects. Figure 3.10 shows the high-level diagram of high speed serial I/O consisting of a serializing transmitter, a channel, and a desterilizing receiver. To operate with a bit period that is small compared to the time-of-flight over the channel, high-speed I/O circuits are typically terminated with a matched impendence at either or both ends to achieve incident-wave signaling and recover the clock phases from the data arriving at the receiver. A major timing noise contributor in high-speed I/O systems is the clock multiplier, which takes a low frequency and in most case, accurate reference clock and synthesizes a high frequency timing reference for the bit stream.

The multiplying DLL architecture is shown in Figure 3.11. This circuit accepts an input clock and generates a phase-locked output clock at a multiple of the input clock frequency. As with a DLL, each rising edge of the input clock zeros the phase error of the loop. Hence this circuit combines the low phase noise of a DLL with the clock multiplication ability of a PLL. Because the same delay elements generate each edge of the output clock in an MDLL, the fixed-pattern jitter due to device mismatch in a conventional DLL frequency synthesizer is eliminated.

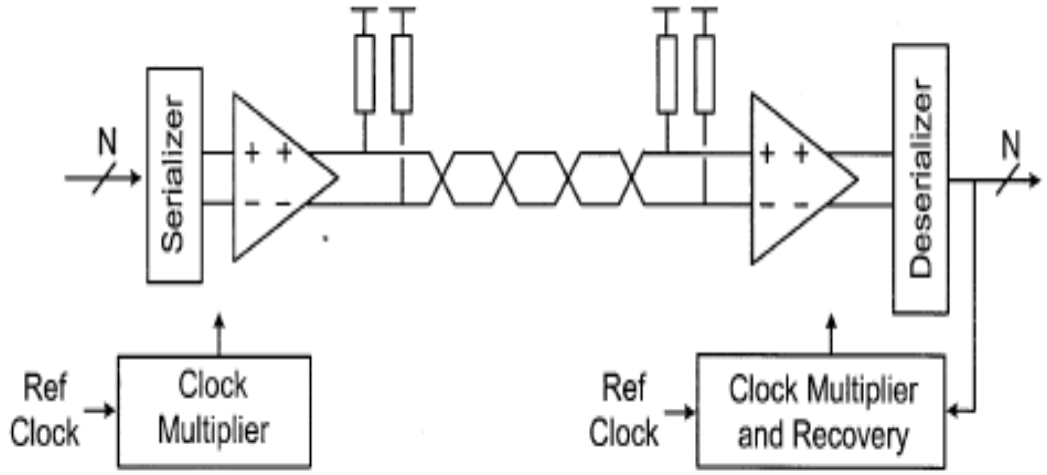


Figure 3.10 Basic components of a high-speed serial I/O

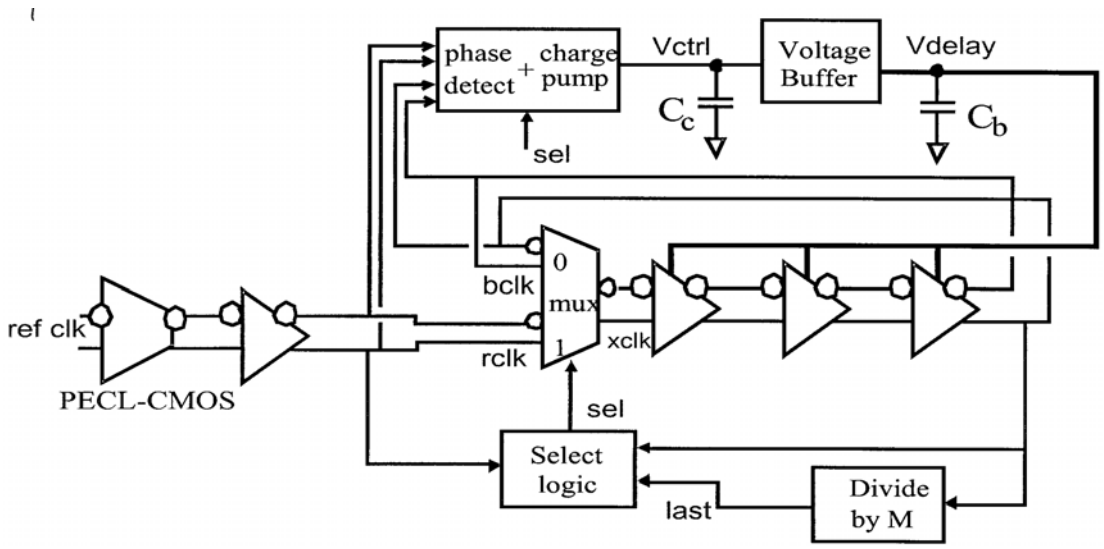


Figure 3.11 Multiplying DLL

3.4.3 Clock Generator for Dynamic Frequency Scaling

In recent years, power consumption has become a critical issue in the embedded systems, especially for the mobile systems. A dynamic Voltage Frequency Scaling (DVFS) becomes more important for saving energy on mobile embedded systems. A DVFS scheme is also proposed in [22]. 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.

Figure 3.12 shows the power consumption comparison between using DVFS and without using DVFS. In MPEG4 application, power consumption reduces 72%; on the other hand, power consumption reduces 83% in PIM application. It is clear that DVFS can save energy significantly.

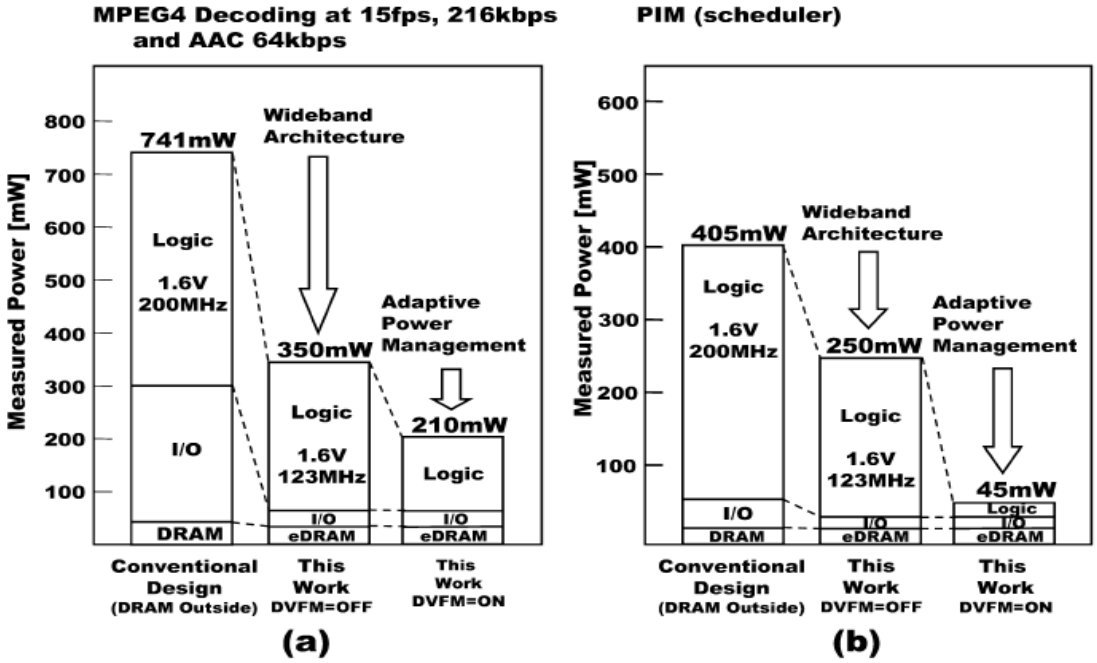


Figure 3.12 Comparison of power consumption

In order not to make performance overhead, the relock time is an important issue for the DFS. A multiphase DLL based clock generator which has fast relock time is proposed in [23]. Figure 3.13 shows the DLL based clock generator block diagram. If the VCDL has N delay cells, then the output frequency can be expressed as equation (1), where $F_{req\_Output\ CLk}$ is frequency of the reference signal and the multiplication factor M/2 can be chosen dynamically by the multiplication factor controller. The digital logic

transition detector and edge combiner is 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.

$$freq\_OutputClk = freq\_Ref \times (M/2), \quad (M = \text{integer}, 1 \leq M \leq N) \quad (1)$$

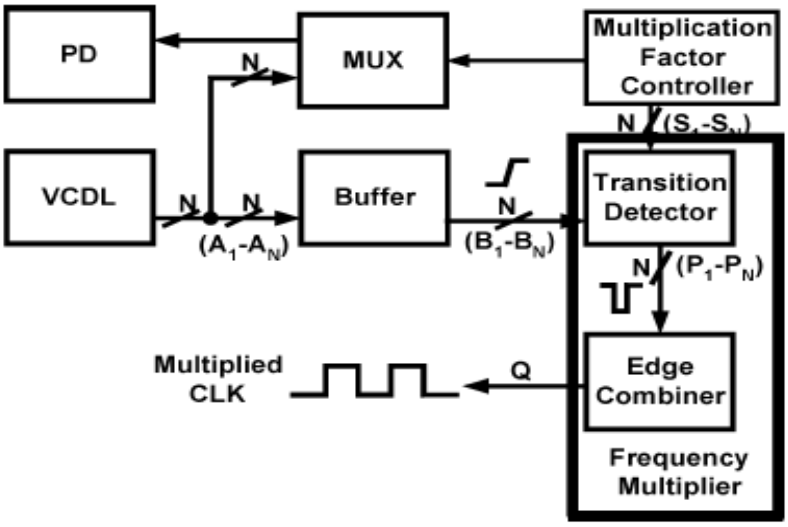


Figure 3.13 Block diagram of DLL-based clock generator

CHAPTER 4
A WIDE POWER SUPPLY RANGE, WIDE LOCKING RANG ALL-DIGITAL MULTIPHASE DLL

In this chapter, an adaptive SAR (ASAR) search algorithm is proposed to achieve fast lock time and extend the locking range. The ASAR search algorithm is based on the frequency-estimation selector and SAR controller, which is designed in digital mode. Compare with the conventional counter-controlled DLL (CDLL), successive approximation register-controlled DLL (SARDLL), and variable SAR DLL (VSARDLL), the ASAR-based DLL can avoid harmonic locking in wide-locking range. Moreover, the ASAR-controller is designed in the robust and low power flip-flop, which is suitable for the wide power supply range operation.

4.1 INTRODUCTION OF WIDE-RANGE DLL

Phase-Locked loops (PLL) and delay-locked loops (DLLs) are widely used as de-skew buffers in microprocessors, memory interface, and communication products. Generally, in several high performance applications, such as double date rate (DDR) SDRAM [17], clock data recovery (CDR) [25], and multi-core processors [26], multiphase DLLs (MDLLs) are often preferred due to their better performances of jitter, stability, and simple design effort than PLLs.

Recently, as the need of wide operating frequency range increasing, conventional MDLLs may suffer from harmonic lock issue. Figure 4.1 shows the harmonic locking

problem [27]. In the DLL, the reference clock, ref_clk, is propagated through VCDL. The output signal, vcdl_clk, at the end of the delay line is compared with the reference input. If delay different from integer multiples of clock period is detected, the closed loop will automatically correct it by changing the delay time of the VCDL. However, the conventional DLL will fail to lock (case 3) or falsely lock to two or more periods (case 4), Tclk, of the input signal if the initial delay of the VCDL is shorter than 0.5*Tclk or longer than 1.5*Tclk. Therefore, if the DLL is required to lock the delay to one clock cycle of the input reference signal, the initial delay of the VCDL needs to be located between 0.5*Tclk and 1.5*Tclk as (case 1 and case 2) shown. As a result, the additional circuit or control mechanism is needed to avoid the false lock. In the next Section, we will introduce the previous research of wide range scheme.

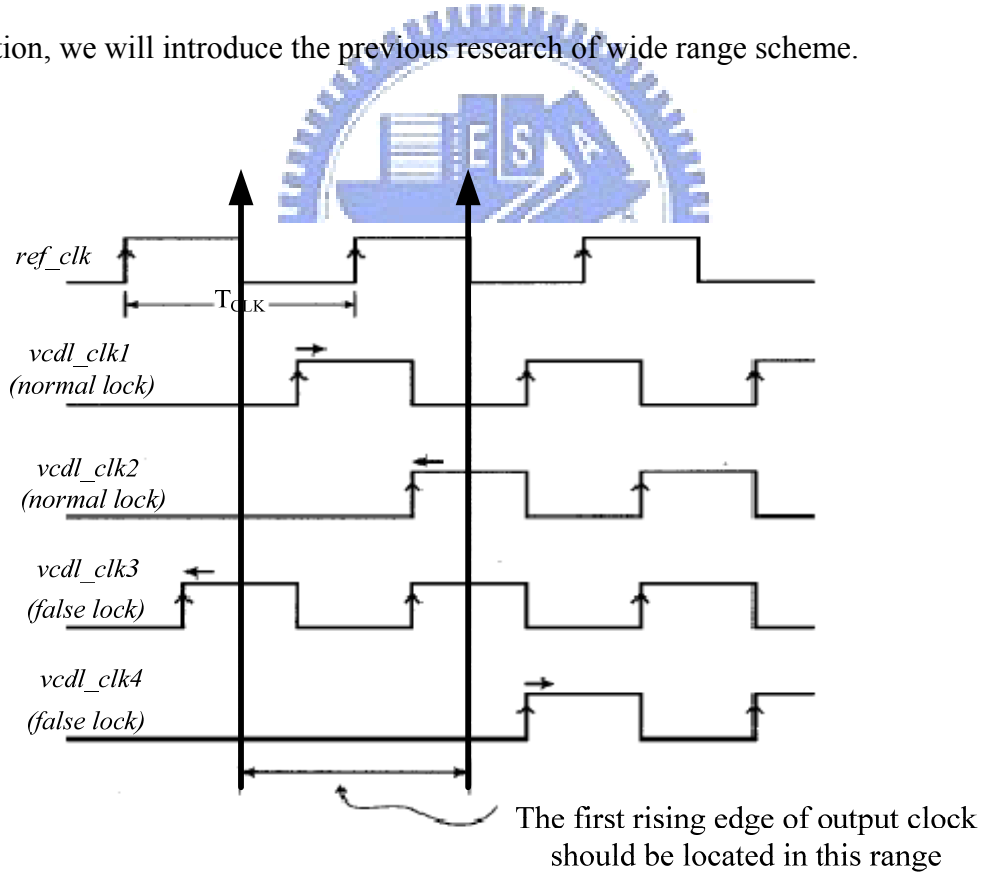


Figure 4.1 Harmonic locking problems.

4.2 PREVIOUS RESEARCH OF WIDE-RANGE DLL

Various wide-range DLLs have been proposed in [8] [28] [29] [30] to solve the harmonic lock problem. In [8], an all-analog DLL uses the replica delay line is proposed to solve the narrow operating frequency range problem of a conventional DLL. The auxiliary loop uses a replica delay line to extend the main loop locking range, as shown in Figure 4.2. If the delay range of the VCDL satisfies the relation $T_{VCDLmin} < 1/7 \times T_{VCDLmax}$, the DLL will have a maximum operation range of 7:1. However, the process-sensitive characteristic makes the analog DLLs difficult to migrate to advanced technologies Therefore, digital DLL is developed to improve process portability.

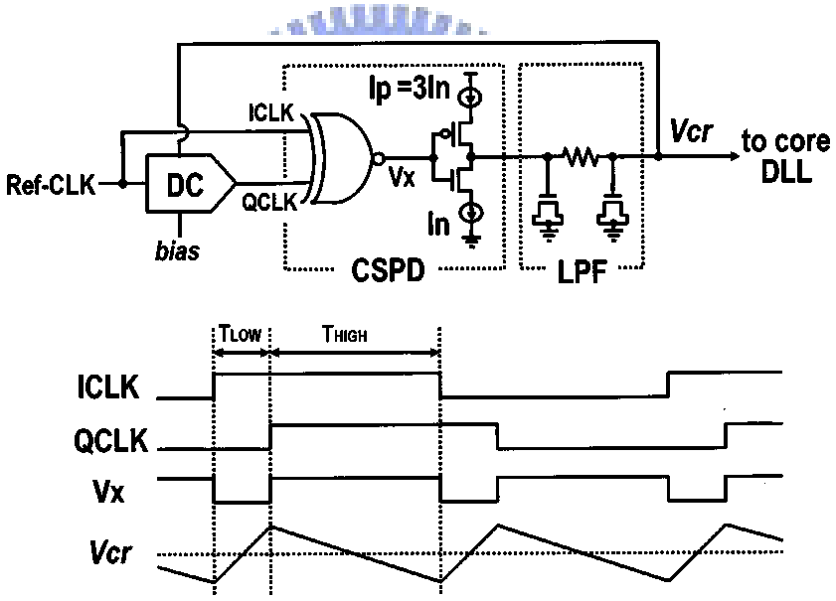


Figure 4.2 Configuration and operation of a replica delay line.

A digital-controlled DLL called the self-correcting DLL is proposed in [30]. The problem of false locking is solved by the addition of a lock-detect circuit and the modified phase detector (PD) as shown Figure 4.3. The modified PD decodes each clock phase and sends over and under signal to avoid harmonic locking, the waveform is shown in Figure 4.4. Although this self-correcting DLL avoids false locking, the outputs of the VCDL are required to have an exact 50% duty cycle.

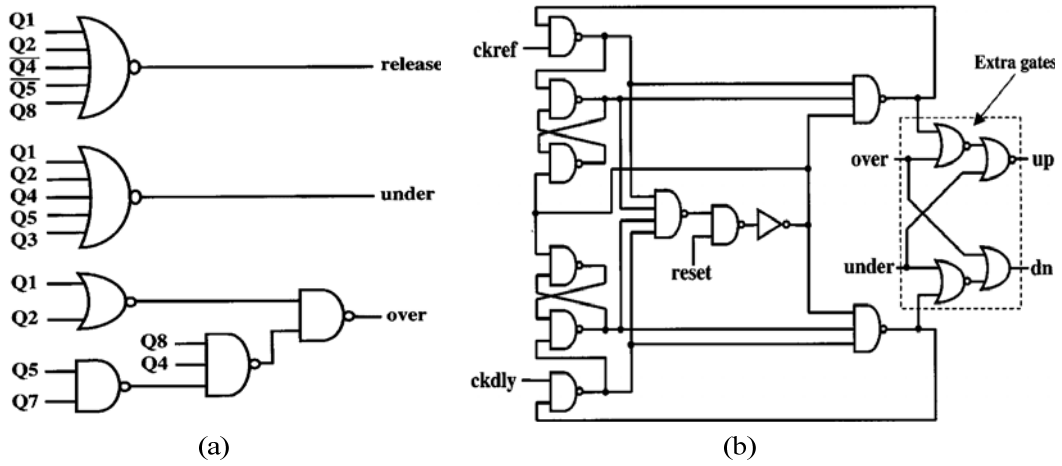


Figure 4.3 (a) Lock-detect decode circuitry (b) Phase detector schematic

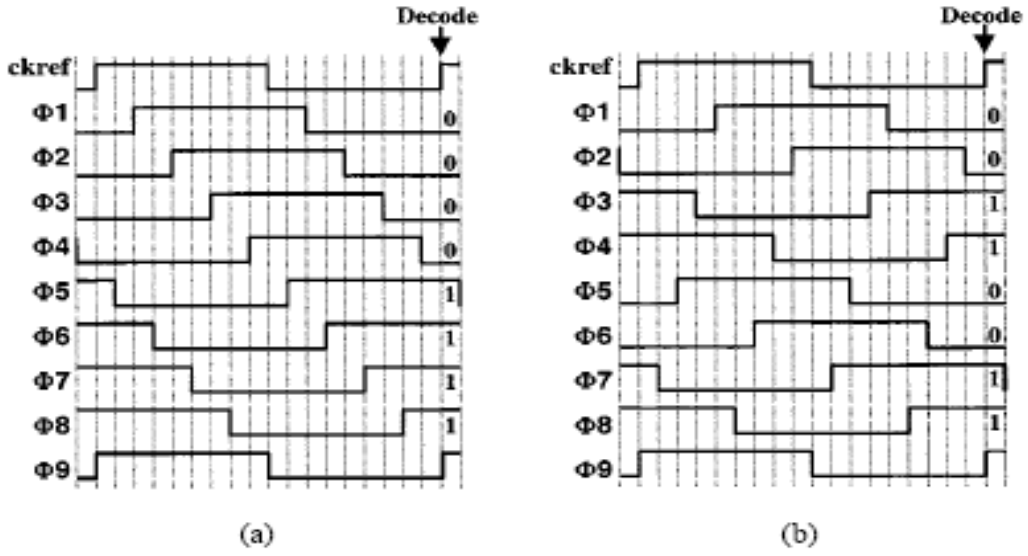


Figure 4.4 Nine-stage VCDL waveforms with (a) correct lock and (b) false lock

The time-to-digital (TDC) scheme may be the simplest concept to resolve the harmonic problem. In [9], the 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 and controls the delay time of the delay line to avoid false locking. However, such DLLs result in complex architectures that face such problems as increased area, increased power consumption. Figure 4.5 shows the simplify TDC architecture.

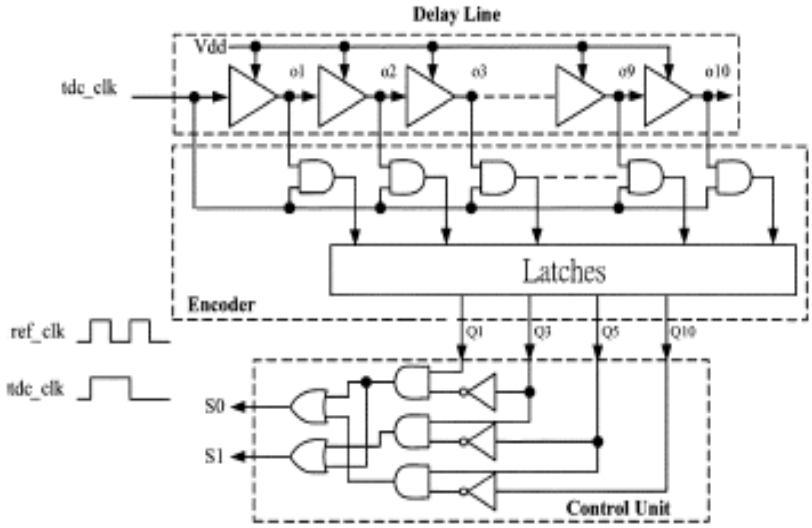


Figure 4.5 The TDC architecture

The variable successive approximation register (VSAR) algorithm is proposed in [8] for all-digital DLL applications. The main 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 form 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. For the variable SAR algorithm, the delay of delay line increases gradually from the minimum and never exceed twice of the input clock period. Therefore, the harmonic locking problem can be avoided. The flowchart of the VSAR algorithm and controller are shown in Figure 4.6

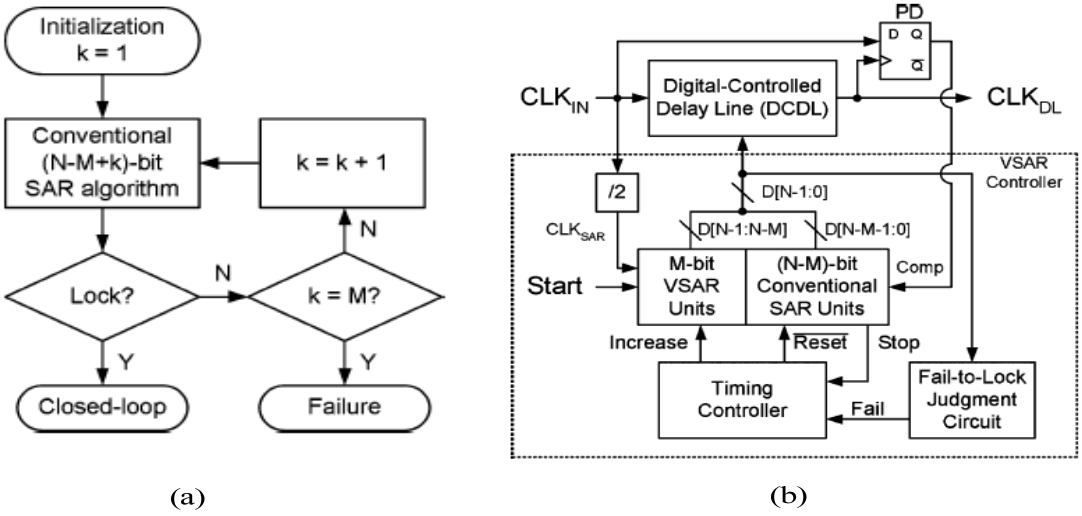


Figure 4.6 (a) The flowchart of the VSAR algorithm (b) VSAR controller

In comparison with the conventional SAR algorithm, the variable SAR algorithm has two advantages: (1) it varies from the open-loop characteristic of conventional binary search algorithm to the close-loop type. (2) The division ratio (DR) based on the variable SAR algorithm can be the minimum of two. Therefore, the variable SAR algorithm can accelerate the locking time and avoid harmonic locking in wide-frequency range. Compare with conventional SAR, up to 7X speedup can be achieved; the simulated lock time versus the operating frequency is shown in Figure 4.7. However, the most drawback of the variable SAR algorithm is still complex hardware required and more power consumption

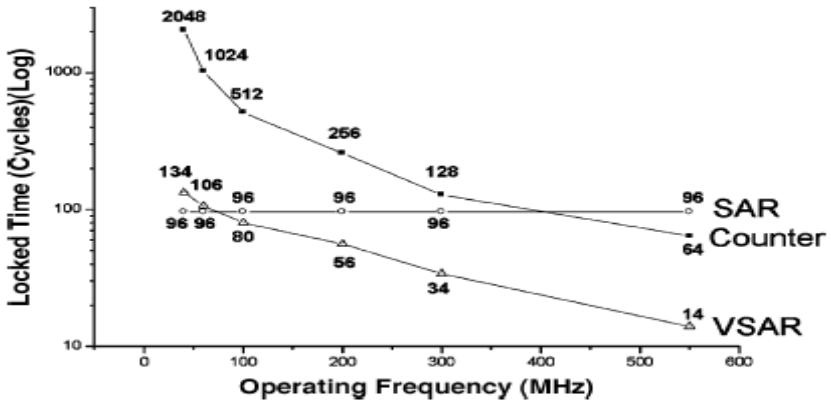


Figure 4.7 Simulated lock time versus the operating frequency.

4.3 ADAPTIVE SAR ALGORITHM

The design challenges of the lock-in controller include lock time, locking range, and area requirement. In order to balance these considerations, a SAR lock-in controller [7] may be the most suitable for all-digital DLLs. However, the conventional SAR controller has harmonic locking issue in wide frequency range, which is not allowed in multiphase DLL applications. To avoid harmonic locking, the delay line range should be always satisfied equation (4.1) [29]

$$Max(T_{DCDL\_MIN}, \frac{2}{3} \times T_{DCDL\_MAX}) < T_{REF} < MIN(T_{DCDL\_MAX}, 2 \times T_{DCDL\_MIN}) \quad (4.1)$$

where T_{REF} means the reference clock period, T_{DCDL_MIN} means minimum delay time of the delay line, and T_{DCDL_MAX} means maximum delay time. For the conventional SAR algorithm, the initial delay time is always set in the middle of delay time as $(T_{DCDL\_MIN}+T_{DCDL\_MAX})/2$. Thus, substitute this condition into equation (4.1), the locking of range conventional SAR algorithm is limited to (4.2).

$$(T_{DCDL\_MAX} + T_{DCDL\_MIN)})/3 < T_{Target} < T_{DCDL\_MAX} \quad (4.2)$$

Equation (4.2) shows that though SAR algorithm has fast locking time, it will fail to lock in wide frequency range without any auxiliary hardware.

The proposed adaptive SAR strategy is using the frequency-estimation selector (the circuit is described in next section) to pre-obtain approximate input frequency range and generates digital codes (S1, S0) to adaptive SAR controller. Meanwhile, the adaptive SAR controller utilizes the codes (S1, S0), giving the multiphase DLL proper initial delay to avoid false locking. For example, if input frequency is high, the LU starts with shorter binary-weighted initial delay, like one-quarter delay time of the delay

line instead of always choosing in the half. In our adaptive SAR controller, we provide three different initial delay of delay line: one-half, one-quarter, and one-eighth, as shown in Figure 4.8. Consequently, the locking range of the proposed adaptive SAR algorithm can be extended as

$$(T_{DCDL\_MAX} + T_{DCDL\_MIN)})/12 < T_{Target} < T_{DCDL\_MAX} \qquad (4.3)$$

From equation (4.3), the proposed adaptive SAR strategy theoretically has maximum locking range when T_{DCDL_MAX} = 11x T_{DCDL_MIN}. Hence, the locking range can be extended from two times to one order, comparing with conventional SAR search. To prevent the false locking caused by PVT variations, the locking range should be overlapped pair-wisely.

Compared with previous research in [1][8][9][29], which require replica delay or complex control circuit, the adaptive SAR strategy has three features: (1) it reuses the delay line to perform time measurement to avoid false lock; therefore, the circuit area and power consumption are significantly reduced; (2) input duty cycle immunity, since it samples two times period of reference clock as frequency measurement. (3) operate in digital mode, which is robust and suitable for wide voltage range operation

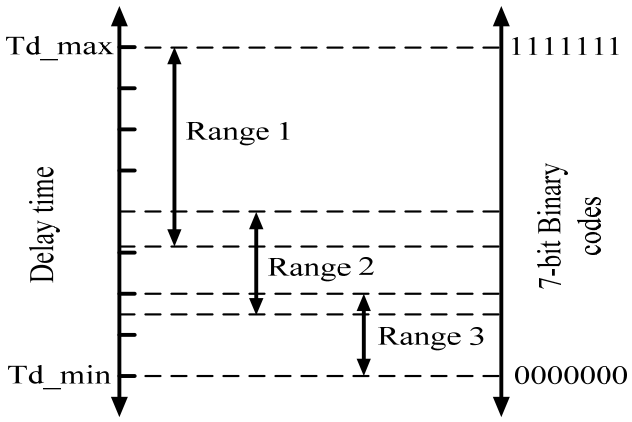


Figure 4.8: The locking range principle of the proposed adaptive SAR.

4.4 CIRCUIT DESCRIPTION

The implementation of adaptive SAR algorithm can simply divide into two parts. They are frequency-estimation selector and adaptive SAR controller. The frequency-estimation selector (FES) would indicate approximate input frequency and generate signals to adaptive SAR controller. Meanwhile, the controller starts with different initial state to perform binary search depend on signals of FES. In following section, we will describe each circuit in detail; moreover, the novel 7-bit resolution four-multiphase DLL is proposed to demonstrate the adaptive SAR algorithm.

4.4.1 Frequency-estimation Selector

The proposed frequency-estimation selector (FES) is shown in Figure 4.9, consisting two D flip-flops. Initially, 7-bit binary-weighted control words C[6:0] are set to "1000000"; therefore, the binary-weighted delay line (BWDL) is in the center of delay range. Among the four phases (P1~P4) of the BWDL, the FES utilizes P2 and P3 to sample CLK_{ref} /2 period and obtain approximately reference clock frequency; meanwhile, the FES generates codes (S1, S0) to the adaptive SAR controller (see in 4.4.2), giving the BWDL proper initial delay. Figure 4.10 shows the timing diagram of the FES. In this work, three different frequency bands are controlled by 2-bit codes (S1, S0), as shown in Figure 4.8 For example, when the reference clock frequency is 1.25GHz, the output of the codes (S1, S0) are (0, 0); the FES is in the "fast" mode, as shown in Figure 4.10. According to Table 1, when the 2-bit control codes (S1, S0) are (0, 0), the adaptive SAR controller starts with binary-weighted initial state "0010000". Consequently, the BWDL initial delay is set as one-eighth of delay time to prevent harmonic locking issue.

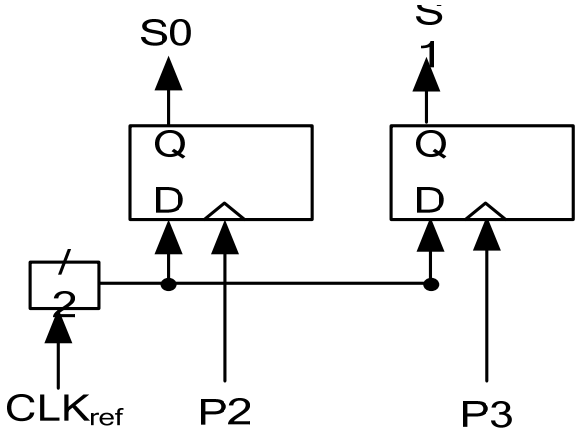


Figure 4.9 The frequency-estimation selector

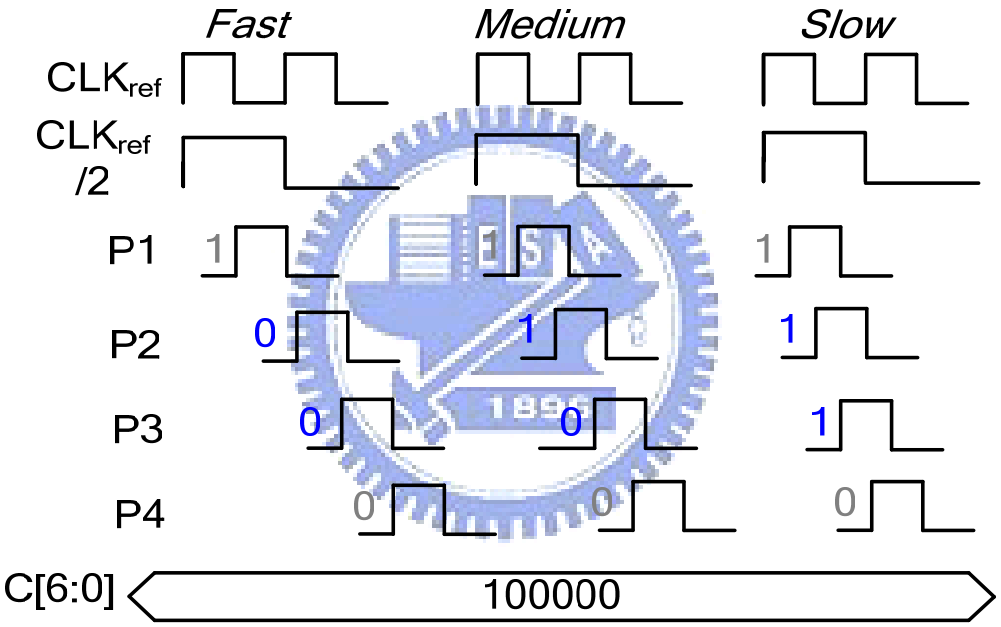


Figure 4.10: The frequency-estimation selector timing diagram

TABLE I. DIGITAL CODES VERSUS INITIAL STATE

Region	P[3:0]	S1, S0	Initial State
Range 1	1110	11	1000000
Range 2	1100	01	0100000
Range 3	1000	00	0010000

4.4.2 Adaptive SAR Controller

The proposed 7-bit adaptive SAR controller is shown in Figure 4.11. It consists of the conventional SAR controller [7] and the adaptive decision block (ADB). The ADB receives control codes (S1, S0) from the FES and generates appropriate digital signals for D6 and D5. If the codes (S1, S0) are (0, 1) or (0, 0), the comparison of SAR search algorithm can further be reduced; hence, the locking steps accelerate as well. The rest of operation steps are basically followed by the SAR search algorithm.

To ensure the adaptive SAR controller can operate accurately in wide freuquency range, the clock signal "clk_sar" is the reference clock cycle divided-by-4. In this work, the LU provides 7-bit resolutions to perform binary search; therefore, the total locking time of the proposed DLL is no more than 28 (7x4) reference clock cycles.

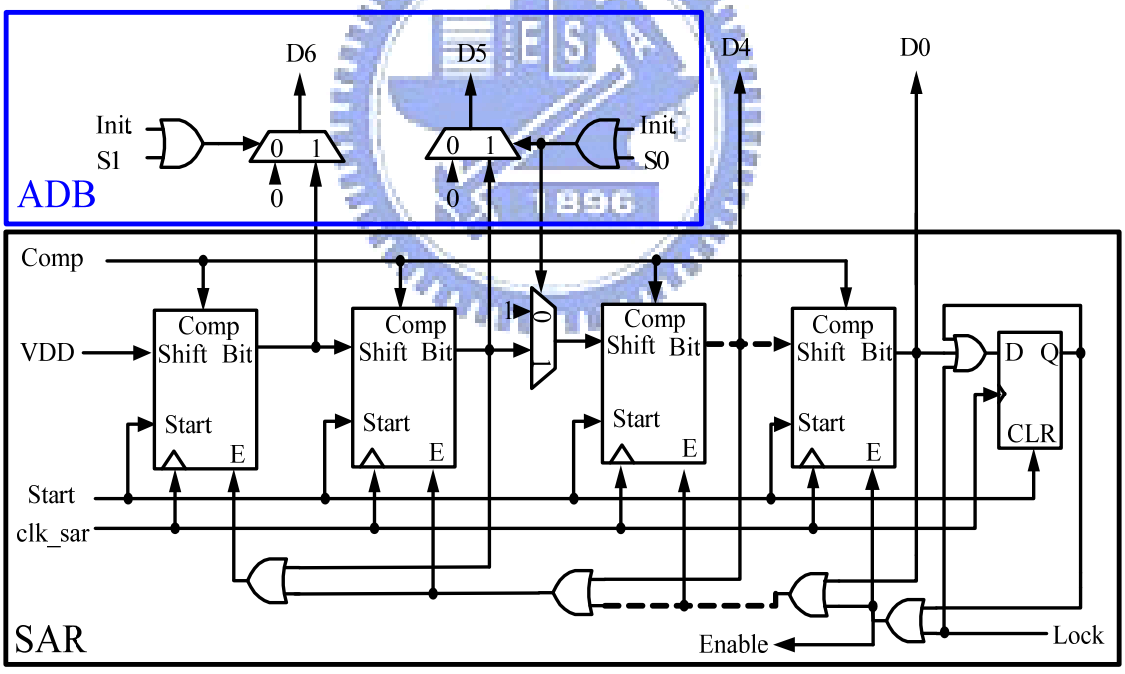


Figure 4.11 7-bit adaptive SAR controller

4.4.3 Digitally Controlled Delay Line

The power consumption, linearity characteristic, delay resolution, and insensitive to PVT variations is main design considerations of the digitally controlled delay line (DCDL). To verify the proposed multiphase DLL, the lattice delay units (LDU) [8] is adopted, as shown in Figure 4.12. The digital control word T0~T2 determines the clock signal (CLKIN) propagation path. Unlike conventional DCDL, which increases tuning range and intrinsic delay at the same time, the LDL intrinsic delay is only two NAND gates. 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. Therefore, as the operating frequency increases, the number of activated delay units is reduced and the power consumption remains the same. To increase the delay resolution, fine tune delay cells similar to [13] are adopted.

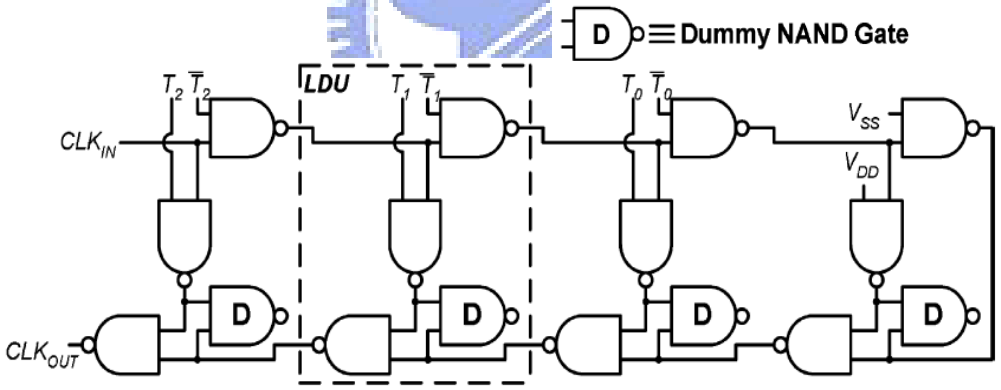


Figure 4.12 lattice delay units

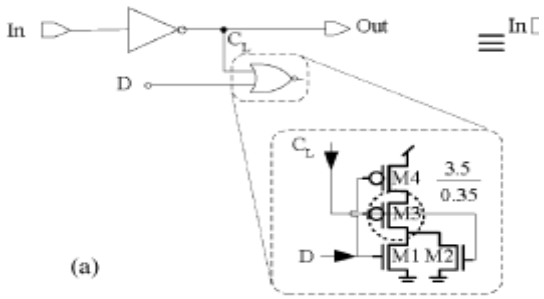


Figure 4.13 Fine tune delay cells

4.4.4 A Robust Ultra-Low Power Design

The adaptive SAR controller accounts for a relatively large portion of the total power consumption; in addition, the major part of the adaptive SAR controller is a flip-flop. In order to operate in wide power supply range, a robust and ultra-low power flip-flop is an important design consideration. In [31], four widely used flip-flops are analyzed, including delay, energy, and energy-delay-product (EDP). Figure 4.14-Figure 4.17 shows these four flip-flop designs: PowerPC master-slaver latch, modified C2MOS (mC2MOS) master-slaver latch, hybrid-latch (HLFF) and sense-amplifier-based (SAFF). PowerPC and mC2MOS are composed of two identical cascaded latches, which are active at different phases of clock signal. HLFF belongs to the class of pulse triggered flip-flop. The input data is latched during a short pulse at the rising edge of the clock. SAFF with two coupled NAND gates as the output latch is a true single clock phase operation.

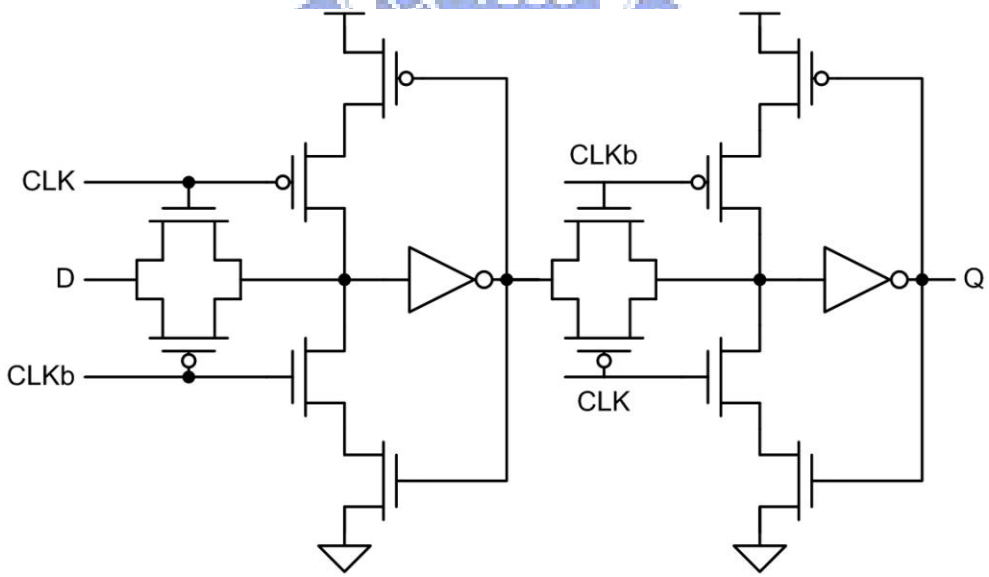


Figure 4.14 PowerPC master-slaver latch (PowerPC).

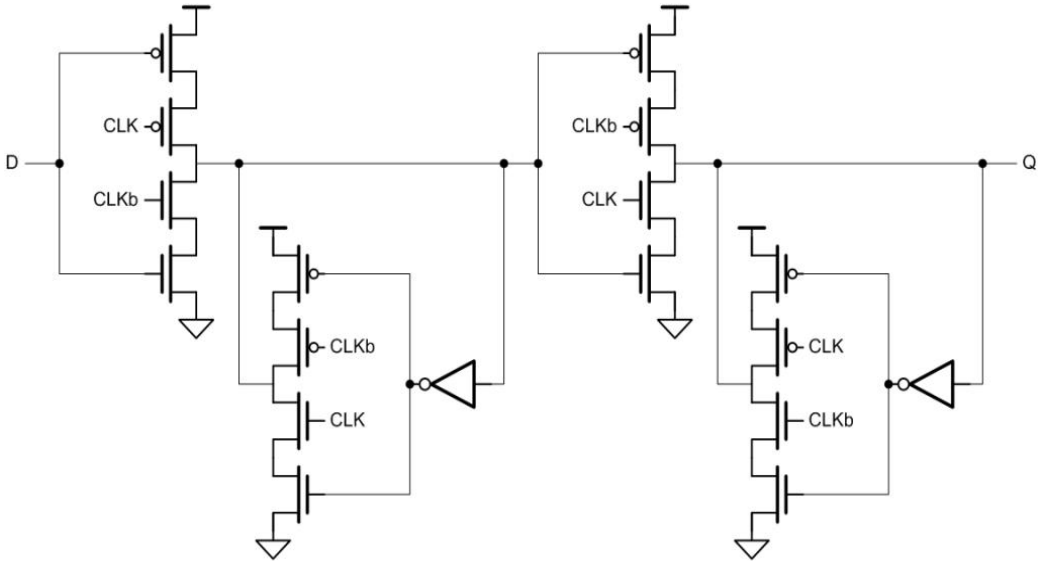


Figure 4.15 Modified C2MOS master-slaver latch (mC2MOS).

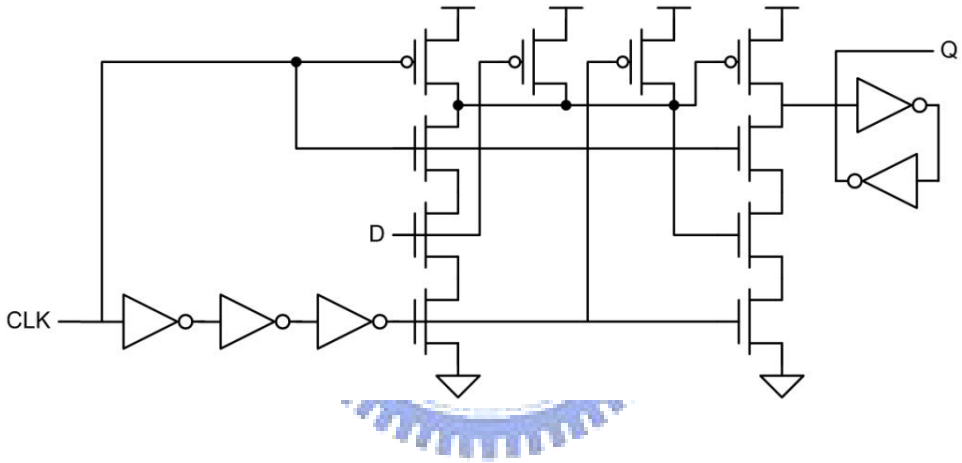


Figure 4.16 Hybrid-latch (HLFF).

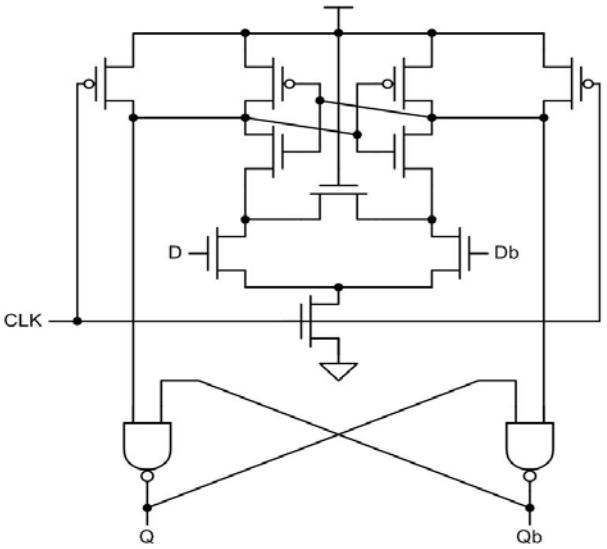


Figure 4.17 Sense-amplifier-based (SAFF).

Figure 4.18 compares the T_{setup}, $T_{C->Q}$ and $T_{D->Q}$ of various flip-flop as the supply voltage is scaled. It is shown that delay time increases as the supply voltage decreases, where delay time increases significantly when supply voltage is in the sub-threshold regime. It is also shown that HLFF achieves the least delay time. This fact is apparent when supply voltage is in the sub-threshold regime. Figure 4.19 shows the energy dissipation of selected flip-flop designs, where PowerPC consumes the least energy. Simulations of energy-delay-product (EDP), an examination vector of the balance between speed and energy consumption, is shown in Figure 4.20 The PowerPC achieves the smallest EDP at high voltages for all switching activities due to its minimal energy consumption and relatively small delay. As supply voltage decreases, the most energy efficient flip-flop architecture depends on switching probabilities, where PowerPC achieves better EDP at low activities, and HLFF achieves better EDP at high activities. Therefore, from the analysis shown above, due to ultra-low voltage operation and low activity of the controller, the PowerPC is chosen to be the basic element.

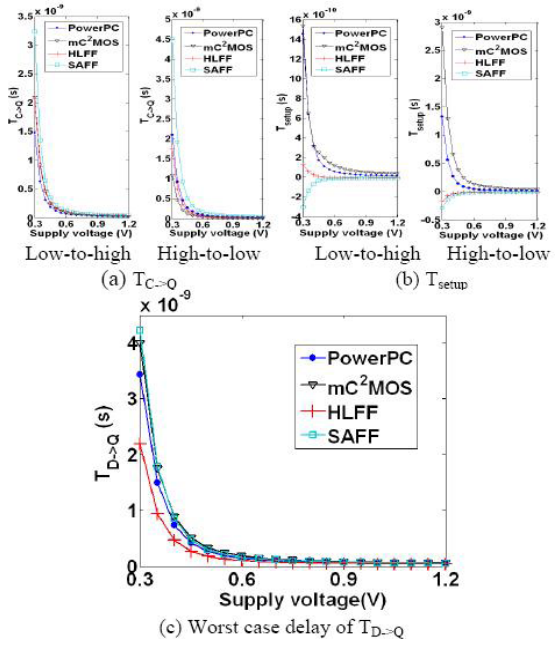


Figure 4.18 Timing parameters of the flip-flop as a function of the supply voltage.

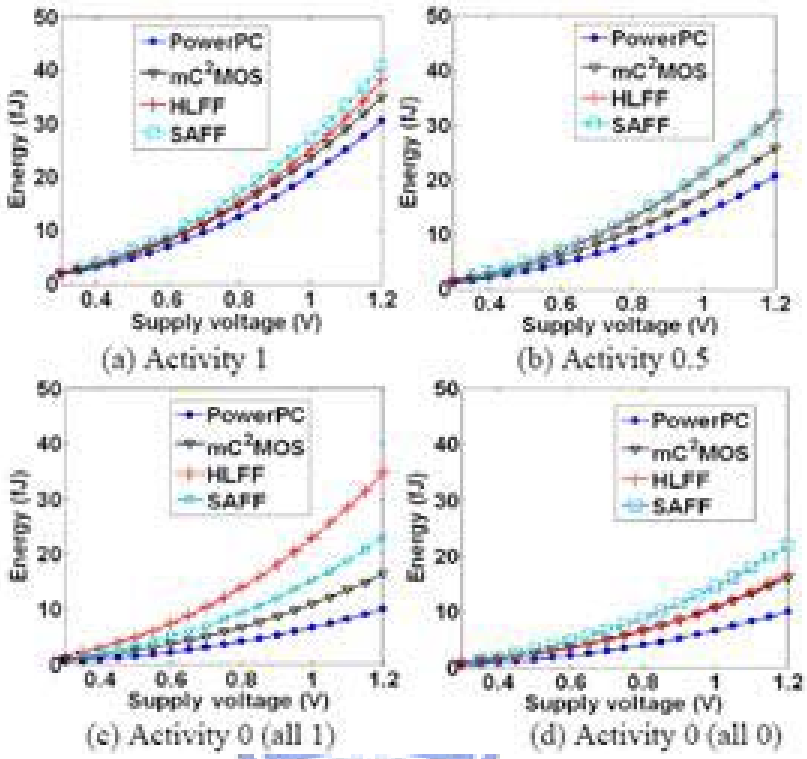


Figure 4.19 Energy dissipation as a function of the supply voltage for different switching

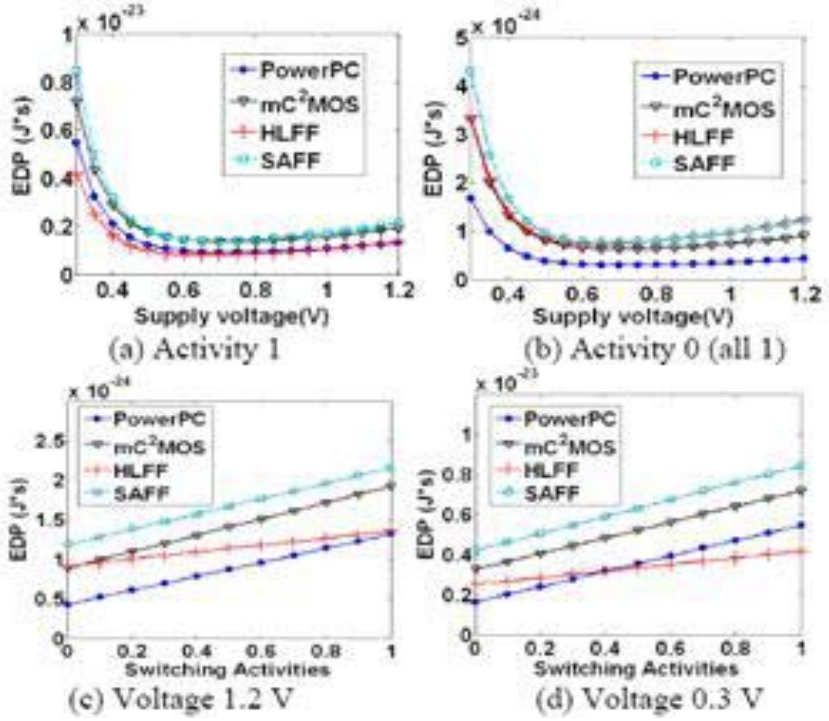


Figure 4.20 EDP as a function of supply voltage and switching activities

4.5 SIMULATION RESULTS

The proposed adaptive SAR algorithm based multiphase DLL can operate from 250MHz to 1.25GHz at 1V and 13MHz to 67MHz at 0.3V based on UMC 90nm CMOS technology. The waveforms show three frequency bands (s1s0=11, 01, 00) at 250MHz, 667MHz, and 1GHz in Figure 4.21-4.23 with 1V, and at 18MHz, 40MHz, and 67MHz in Figure 4.24-4.26 with 0.3V. The simulation result shows the DLL is functional within all the frequency and all the supply voltage. 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) *S1*: The frequency select signal.

(5) *S0*: The frequency select signal.

(6) *Comp*: The output of phase detector. If *LEAD* is 1, increase the delay; otherwise, decrease the delay.

(7) *Lock*: The lock signal

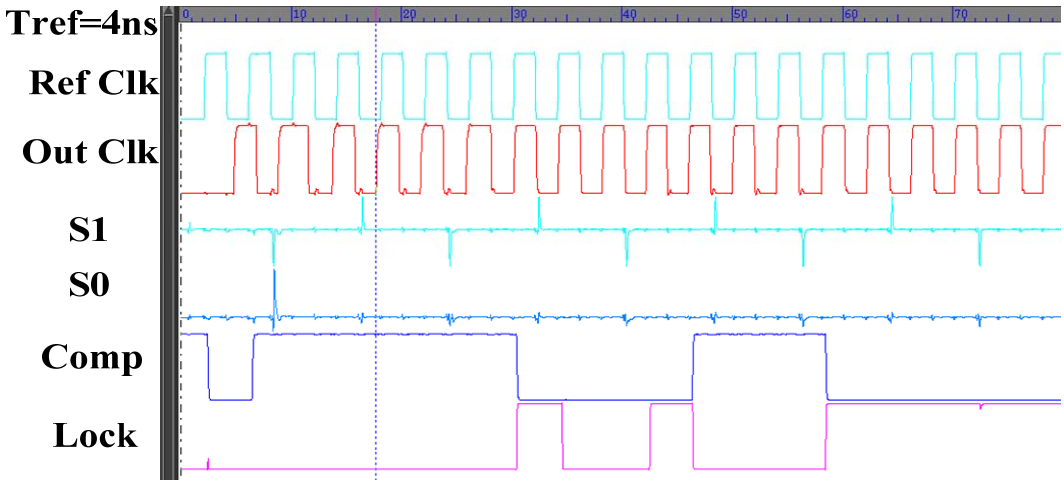


Figure 4.21 Lock process when the input frequency is 250MHz at1V (S1S0=11)

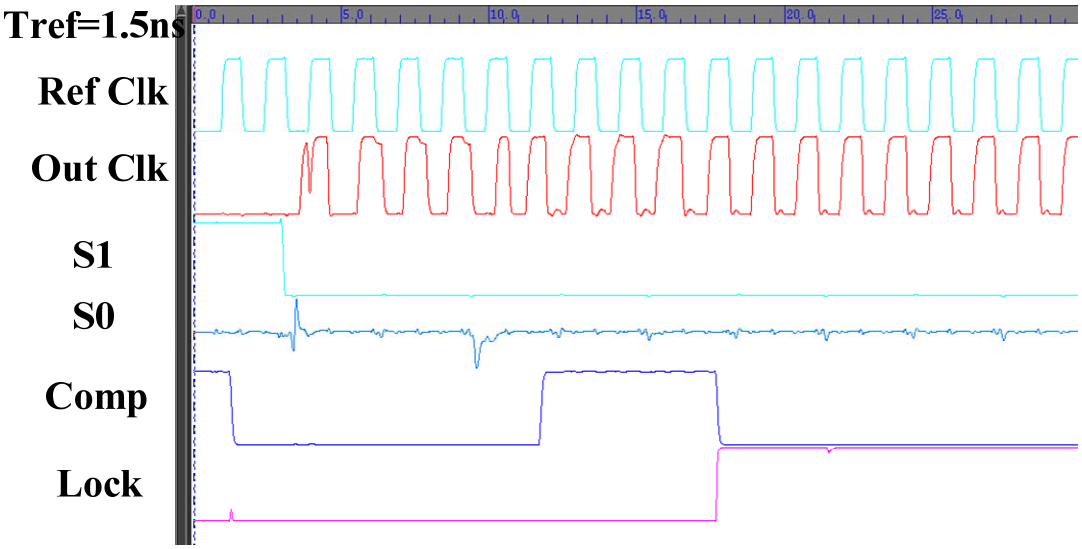


Figure 4.22 Lock process when the input frequency is 667MHz at1V (S1S0=01)

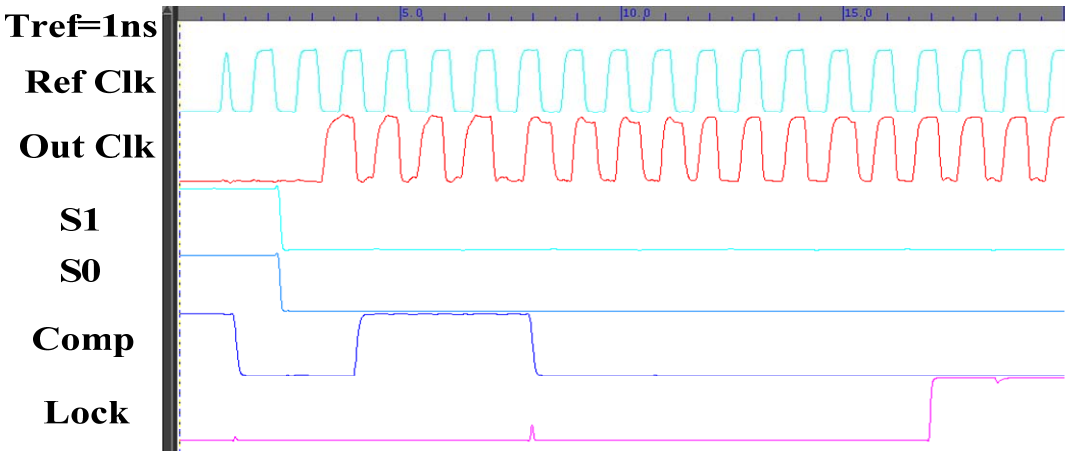


Figure 4.23 Lock process when the input frequency is 1GHz at1V (S1S0=00)

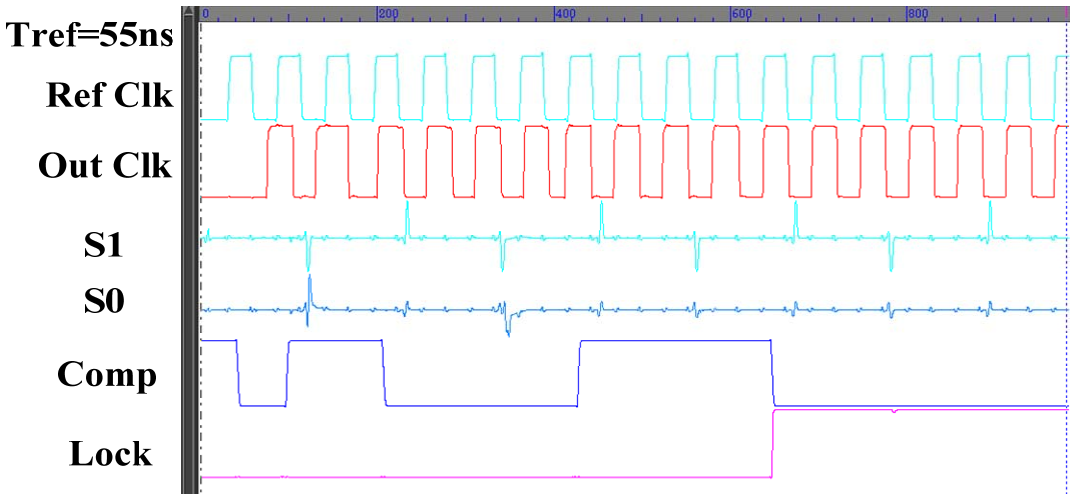


Figure 4.24 Lock process when the input frequency is 18 MHz at 0.3V (S1S0=11)

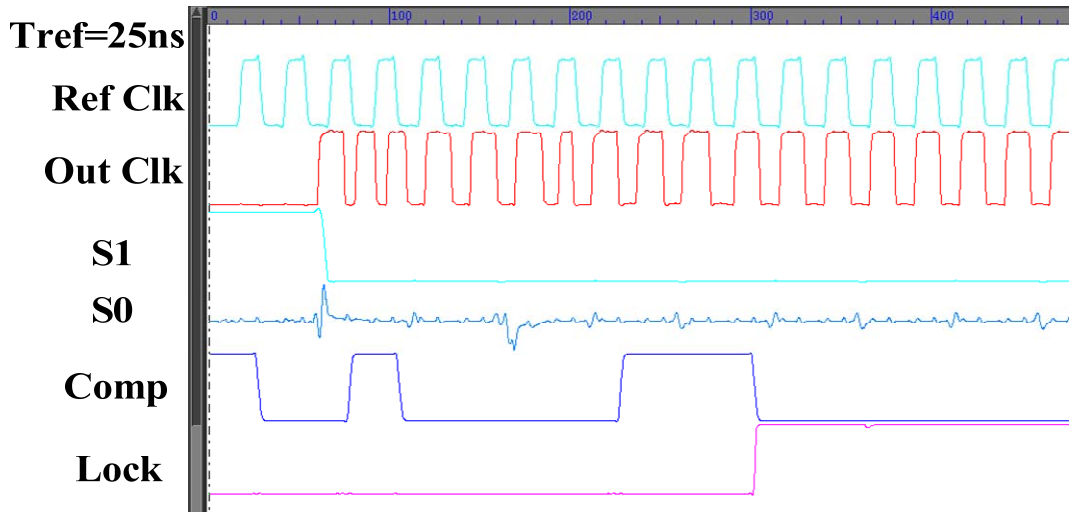


Figure 4.25 Lock process when the input frequency is 40 MHz at 0.3V (S1S0=01)

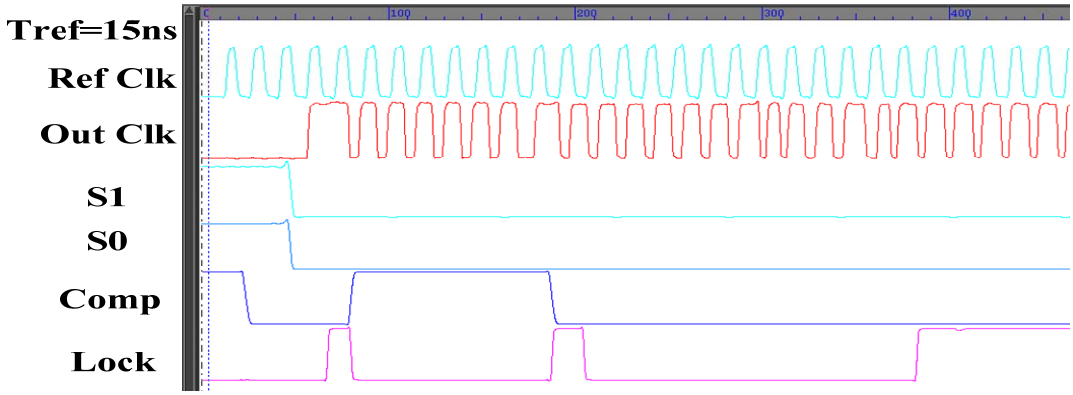


Figure 4.26 Lock process when the input frequency is 67 MHz at 0.3V (S1S0=00)

CHAPTER 5
IMPLEMENTATION OF ALL-DIGITAL PROGRAMMABLE DLL-BASED FREQUENCY MULTIPLIER

In this chapter, a 125MHz-2.5GHz all-digital programmable DLL-based frequency multiplier (ADP-DLLFM) has been designed in UMC 90nm CMOS technology model. A proposed adaptive SAR controller can accelerate the locking procedure and provide a wider locking range. After the ADP-DLLFM is locked, the dynamic frequency monitor circuit is enabled to compensate the phase error caused by PVT variations. We also propose the leakage-reduced delay unit which has low leakage power consumption and insensitive to PVT variations. Finally, we present the frequency multiplier takes advantages of duty cycle immunity, low power consumption, and robust in wide supply voltage range.

An overview of the proposed ADP-DLLFM is introduced in Section 5.1. Section 5.2 presents the circuit implementation to fulfill the intention of the proposed ADP-DLLFM. The simulation results of ADP-DLLFM which is implemented in UMC 90nm CMOS technology model are shown in Section 5.3. Finally, conclusion and simulation results are given in Section 5.4.

5.1 SYSTEM ARCHITECTURE

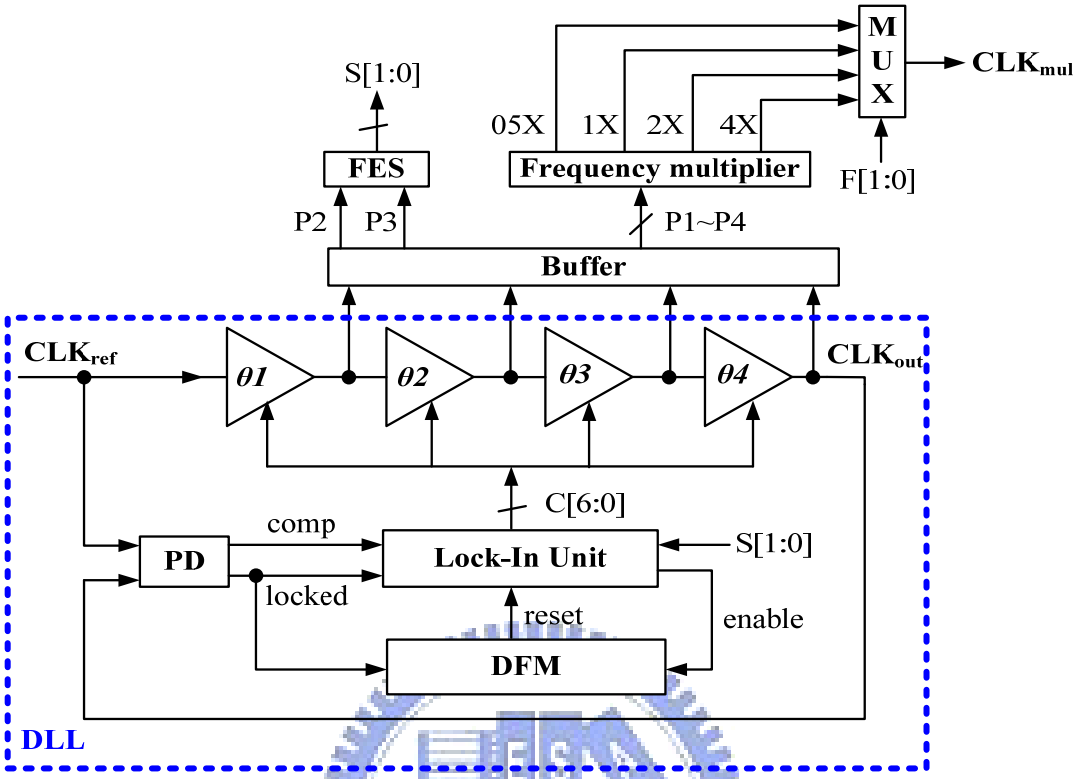


Figure 5.1: The proposed all-digital DLL-based frequency multiplier architecture

The proposed all-digital programmable DLL-based frequency multiplier (ADP-DLLFM) is shown in Figure 5.1. It consists of six major blocks; they are digitally controlled delay line (DCDL), phase detector (PD), frequency-estimation selector (FES), lock-in unit (LU), dynamic frequency monitor (DFM) unit, and the frequency multiplier. The entire operation includes three steps: the period measurement, the synchronizing, and the frequency variation compensation. In the period measurement step, the LU activates the FES and sets the DCDL in the middle point of delay range. Then, the FES utilizes phase P2 and P3 of DCDL to estimate approximate input frequency range and generate digital codes S[1:0]. In the synchronizing step, the LU uses digital codes S[1:0] and PD to generate binary-weighted control word C[6:0] to adjust delay time of DCDL . Once the DCDL is aligned with the reference clock and delay in one reference clock cycle, the

ADP-DLLFM is in the lock state. Finally, the ADP-DLLFM enters the frequency compensation step, the LU enables the DFM circuit to track dynamic phase error caused by PVT variations. When the phase error between CLK_{ref} and CLK_{out} is out of the locking window, the DFM generates the internal reset signal to the LU and adjusts the phase error. Total compensation cycles are no more than 28 reference clock cycles.

When the ADP-DLLFM is locked, the output phases of DCDL are evenly spaced one reference clock period $T_{ref.}$. Then, each phase is fed into the proposed frequency multiplier to generate 0.5X, 1X, 2X, and 4X of reference frequency. The output frequency of CLK_{mul} can be choosen depending on different frequency selector signal F[1:0] .. In the next section, we will describe each circuit in detail.

5.2 CIRCUIT DESCRIPTION

5.2.1 Phase Detector

The phase resolution and the operating speed are key design parameters of the phase detector (PD). The conventional PD is constructed with two dynamic latches (TSPC) [7] can achieve high speed and high resolution for the DLL application. Unfortunately, the TSPC-based PD is easy to malfunction in ultra-low voltage operation; therefore, the robust PowerPC-based PD is proposed to operate correctly in wide supply voltage range.

The phase detector (PD) consists of two PowerPC Flip-Flops (DFFs) and some logic circuits, as shown in figure 5.2. The PD uses these two PowerPC DFFs and buffers to forming the lock detecting window *2Td*. In this design, the *2Td* is equal to 30ps. When the feedback clock is located inside the window, the *Lock* signal rises. On the other hand, if the feedback clock leads or lags reference clock, the *Comp* is 1 or 0. Figure 5.3 shows operation principle of PD.

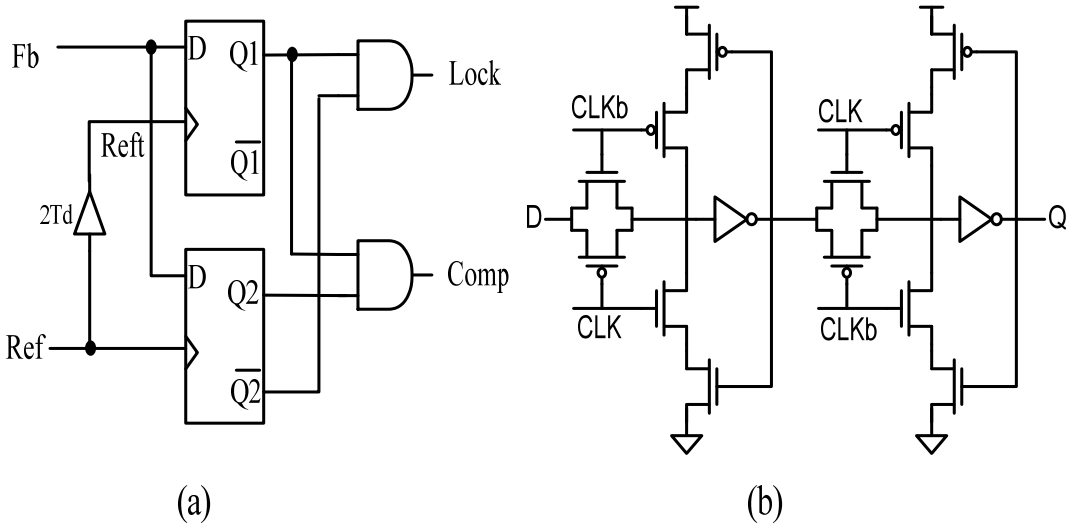


Figure 5.2 (a) The block diagram of phase detector (b) PowerPC flip-flop

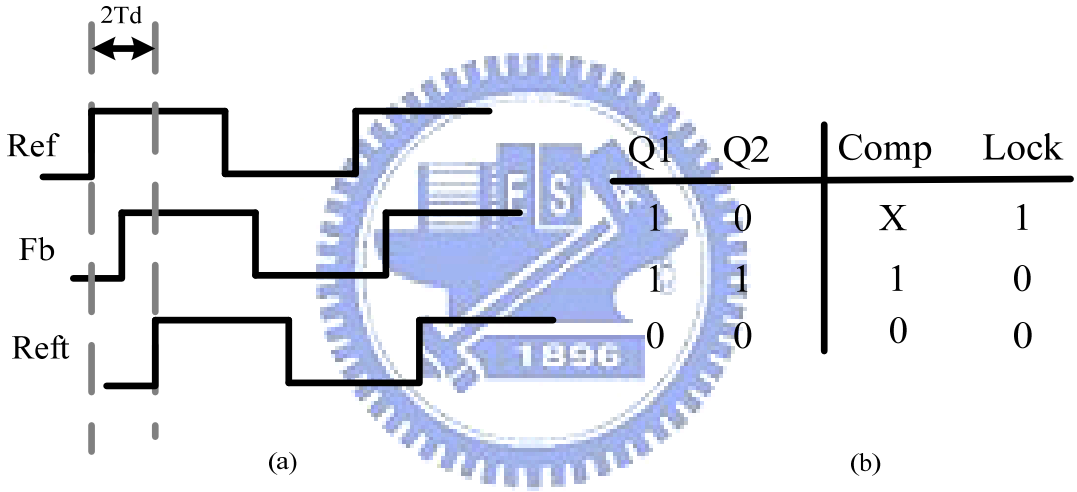


Figure 5.3 Operation principle of proposed PD

5.2.2 Leakage-reduced Delay Line

Before starting to describe the proposed leakage-reduced delay line, we briefly describe several types of leakage current; moreover, we will introduce transistor stacking technique, which can mitigate leakage current. There are four main sources of leakage current in a CMOS transistor as illustrated in Figure 5.4[32]. They are reverse-biased junction leakage current (I_{REV}), gate induced drain leakage (I_{GIDL}), gate direct-tunneling leakage (I_G), and subthreshold leakage (I_{SUB}). Each source of leakage

current will be described clearly in [32].

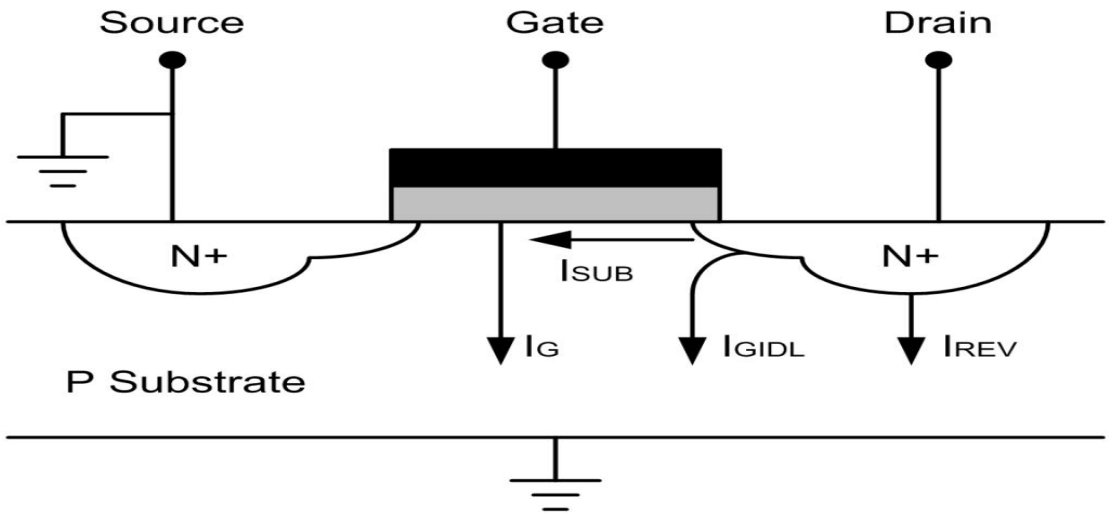


Figure 5.4 Leakage current components in an NMOS transistor.

Transistor stacking is an effective technique to reduce subthreshold and gate leakage current. Leakage current owing through a stack of series-connected transistors reduces if more than one transistor in the stack is one, which is known as the stacking effect. The staking effect can be understood by considering a two-input NAND gate, as shown in Figure 5.5 When both MN1 and MN2 are off, the voltage at the intermediate node (VM) raises to a positive value due to a small drain current. Positive potential at the intermediate node leads to three effects:

1. Gate-to-source voltage of MN1 becomes negative.

2. Negative body-to-source potential of MN1 causes more body effect. The body effect describes how the potential difference between source and body affects the threshold voltage, which can be modeled as:

$$V_T = V_{T0} + \gamma\left(\sqrt{\Phi_S + V_{SB}} - \sqrt{\Phi_S}\right) \quad (5.1)$$

3. Drain-to-source potential of MN1 decreases, resulting in less drain-induced barrier lowering.

As a result, negative gate-to-source voltage, higher threshold voltage due to the

body effect, and less drain-induced barrier lowering due to the reduction of drain-to-source voltage, leakage current is reduced.

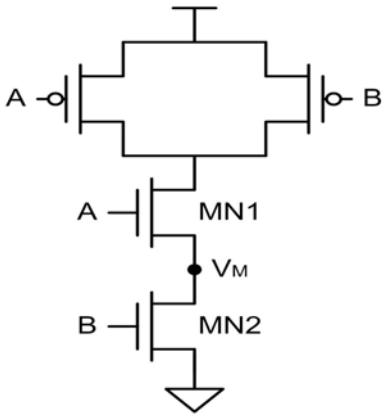


Figure 5.5 Two-input NAND gate stacking effect illustration.

Figure 5.6 show the two-input NAND gate input vector versus leakage power, and the simulation is based on UMC 90nm model. It is clearly that when two input vector AB is equal to 11, there is the most leakage power. Hence, the half-stack NAND gate is proposed as the key component of leakage-reduced delay line to mitigate leakage power when input vector is equal to 11.

Figure 5.7 shows the conventional NAND and the proposed half-stack NAND gate (HSND). When two input vectors are equal to 11, the HSND can mitigate half of leakage power due to the stack effect introduced above. Figure 5.8 illustrate the standby power comparison between NAND and HSND, while Figure 5.9 is the active power. Time delay comparisons are shown in Figure 5.10. From the Figure 5.8 and Figure 5.10, it is obvious that the proposed HSND can reduce leakage power significantly, most important, it is nearly not degrading the speed performance of delay line which is the key factor for clock generator.

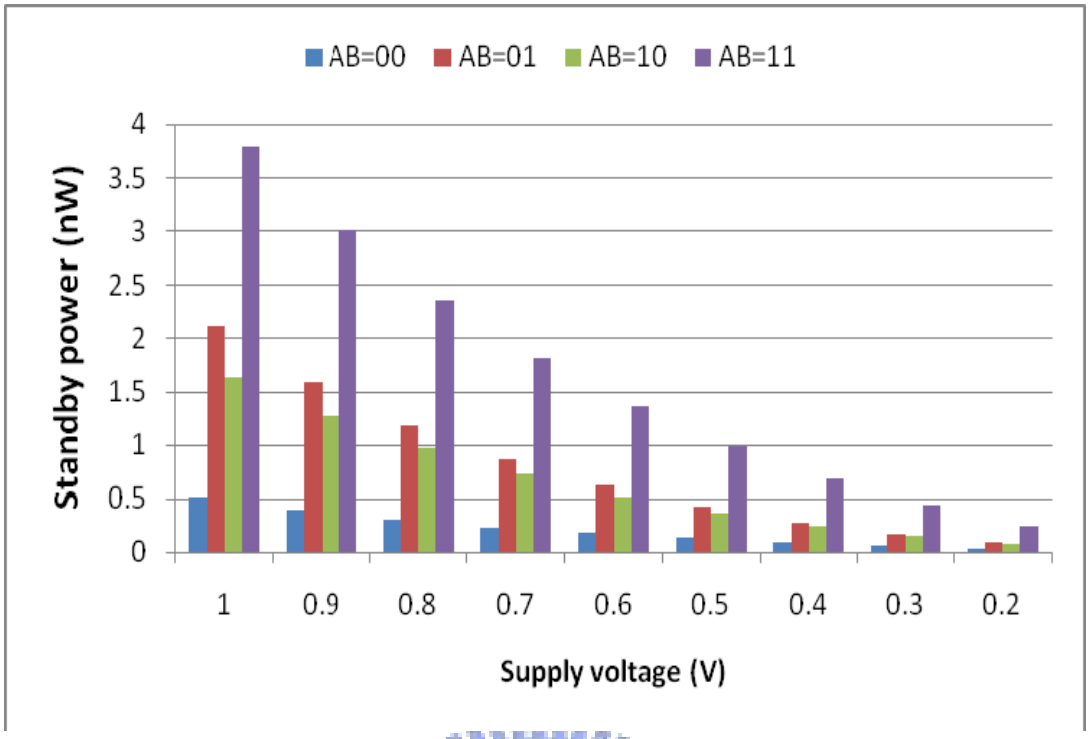


Figure 5.6: Input vector versus leakage current.

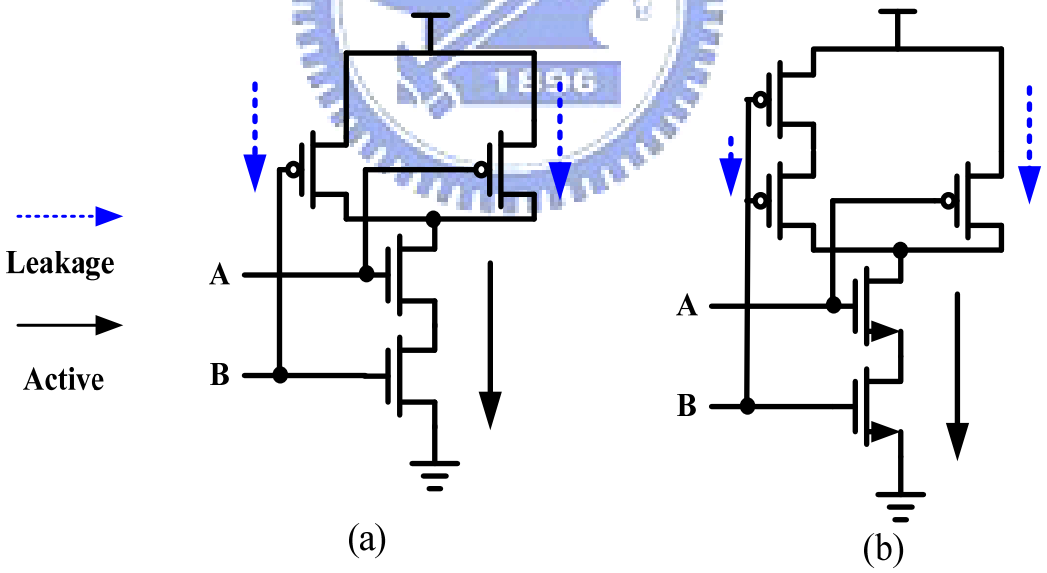


Figure 5.7: (a) NAND (b) Half-stack NAND

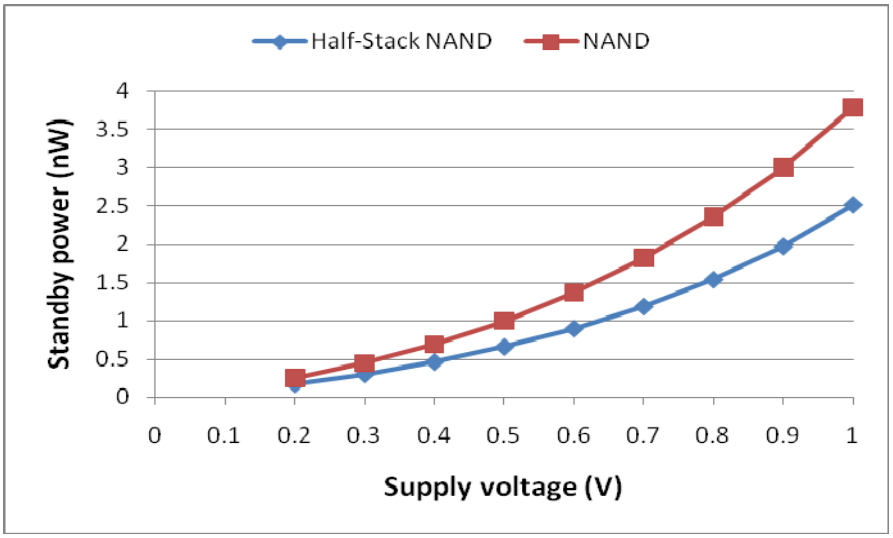


Figure 5.8 Standby power comparisons

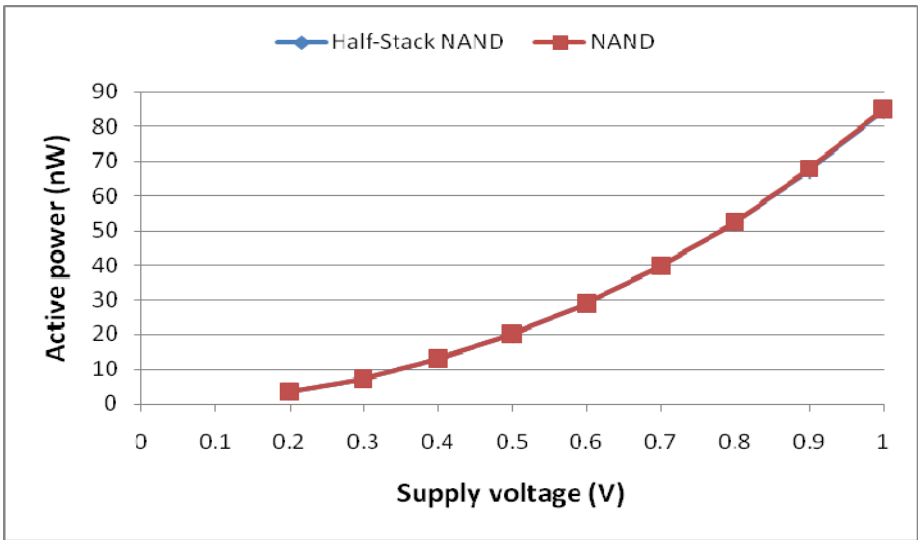


Figure 5.9 Active power comparisons

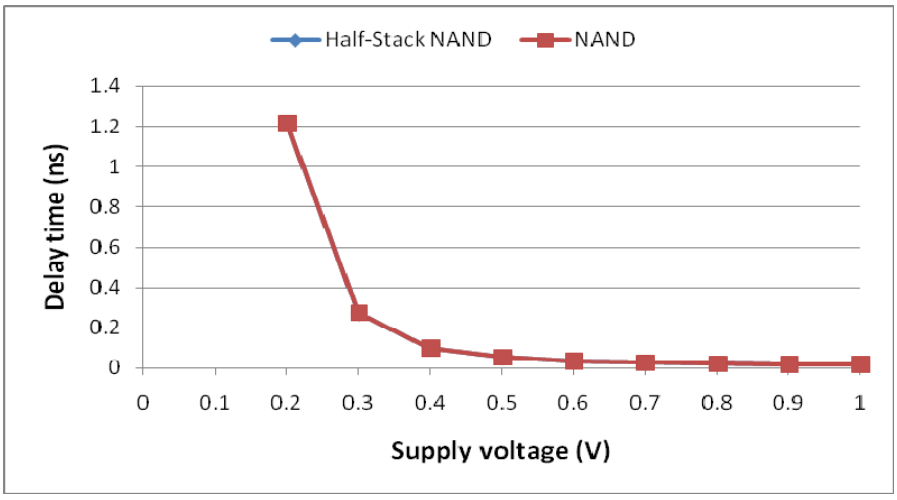
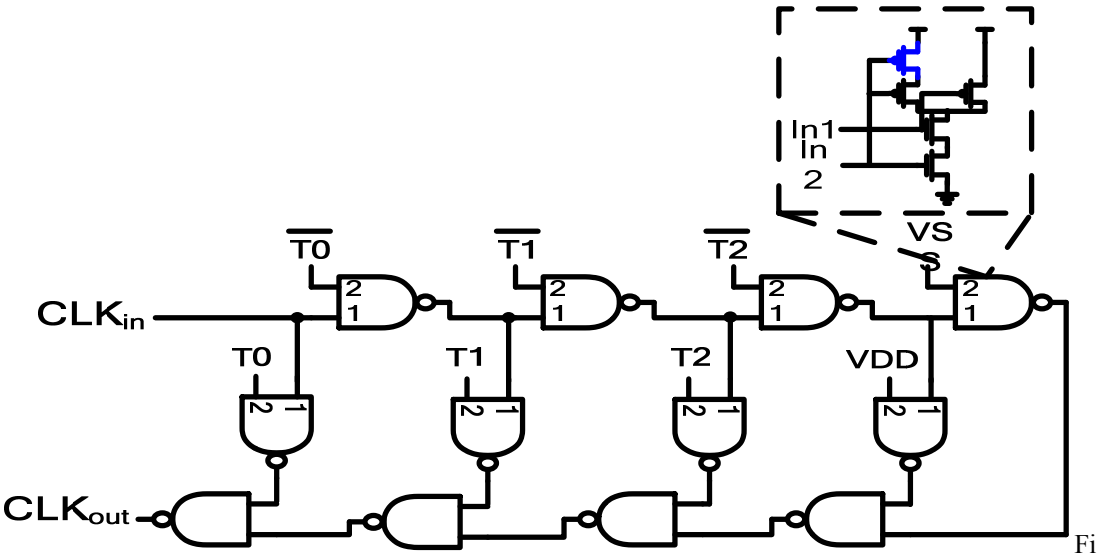


Figure 5.10 Time delay comparisons

The power consumption, linearity characteristic, delay resolution, and tolerance to PVT variations are the main design considerations of the digitally controlled delay line (DCDL). Therefore, the leakage-reduced delay line (LRDL) is proposed to achieve low power consumption, linearity, and robust in wide voltage range, as shown in Figure 5.11. It consists of two different types of NAND gates. One is a conventional NAND gate; the other is a half-stack NAND gate introduced in page 69. To have a better understanding of LRDU, Figure 5.11 takes only three digital codes T0~T2 for example; however, in the proposed clock generator, there are T0~T15 codes to achieve wide delay range.

The binary-weighted control words are converted into the thermometer codes T0~T2. The digital control word T0~T2 determines the clock signal (CLKIN) propagation path. Unlike conventional DCDL, which increases tuning range and intrinsic delay at the same time, the LRDL intrinsic delay is only two NAND gates. When the tuning range increases, the minimal delay is not changed. Both the intrinsic delay and the delay step in an LRDL are the delay of two NAND gates. Therefore, as the operating frequency increases, the number of activated delay units is reduced and the power consumption remains the same. Moreover, the leakage power mitigates 10% due to stack effect of the proposed half-stack NAND gate. To increase the delay resolution, the fine tune delay unit similar to [13] is added in front of the LRDU. Figure 5.12 and show Figure 5.13 shows comparisons of standby power and delay time.



Fi

Figure 5.11: The proposed leakage-reduced delay line

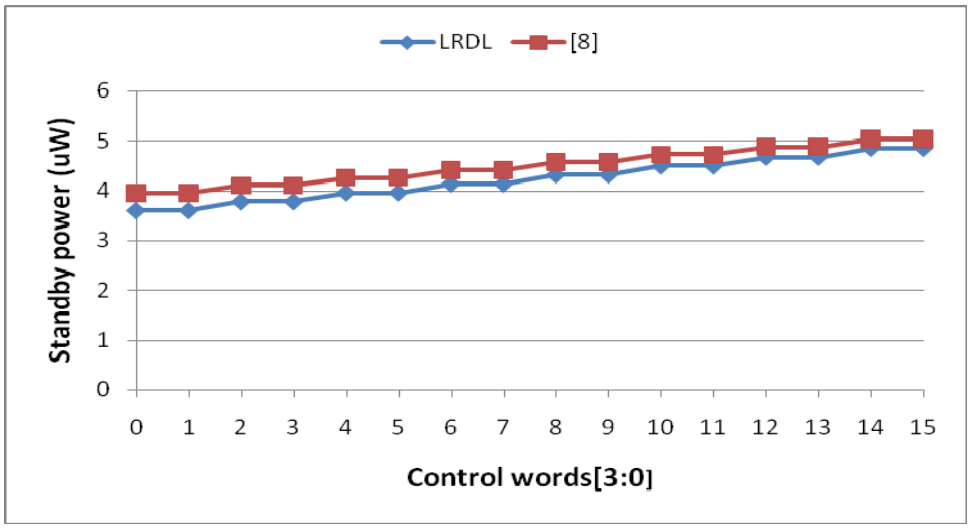


Figure 5.12: Standby power comparisons

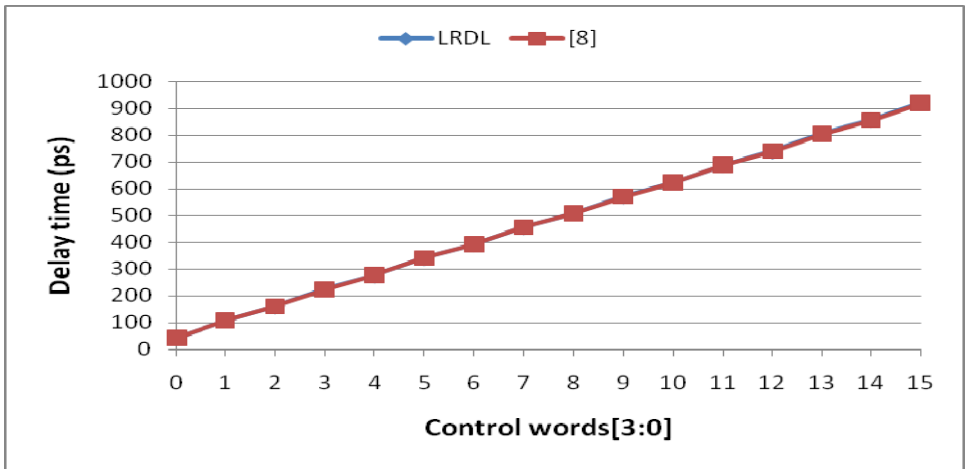


Figure 5.13: Time delay comparisons

5.2.3 Lock-in Unit

The architecture of lock-in unit is based on the adaptive algorithm as we mentioned in Section 4.3. Figure 5.14 shows the 7-bit lock-unit which is composed of the adaptive decision block (ADB) and conventional SAR controller. The ADB receives control codes S[1:0] from the FES and generates appropriate digital signals for D6 and D5. If the codes S[1:0] are equal to (0, 1) or (0, 0), the comparison of SAR search algorithm can further be reduced; hence, the locking steps accelerate as well. The rest of operation steps are basically followed by the SAR search algorithm in [7].

To ensure the LU can operate accurately in wide-locking range, the LU clock signal CLKsar is the reference clock cycle divided-by-4. In this work, the LU provides 7-bit resolutions to perform binary search; therefore, the total locking time of the proposed DLL is no more than 28 (7x4) reference clock cycles. Figure 5.15 shows the simulation result of the lock-in unit when the operating frequency is 1GHz at 1V.

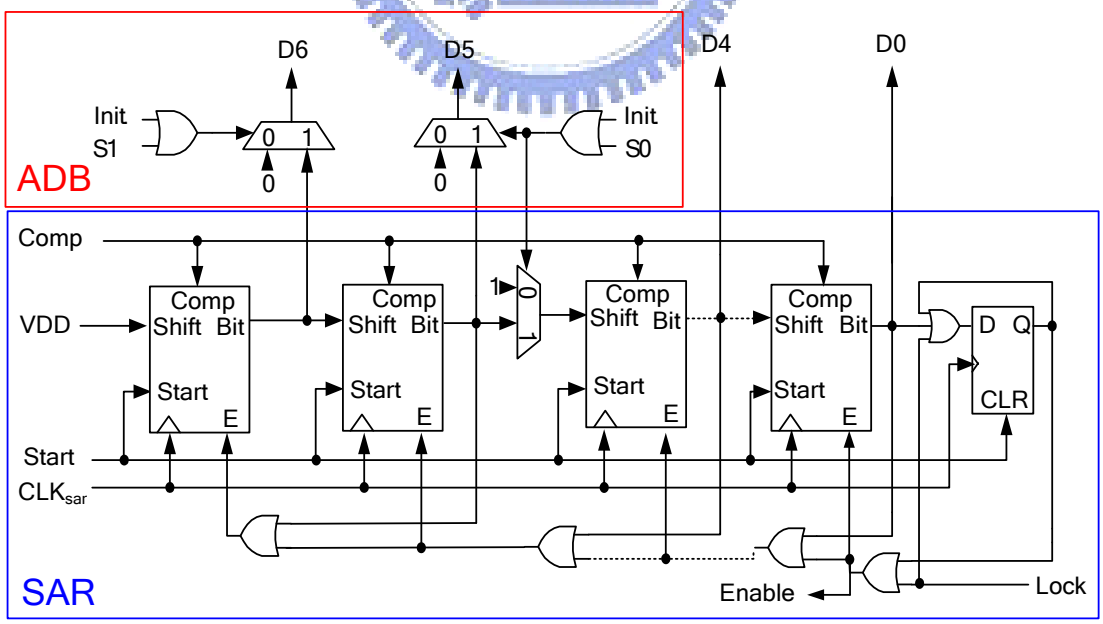


Figure 5.14: 7-bit lock-in unit

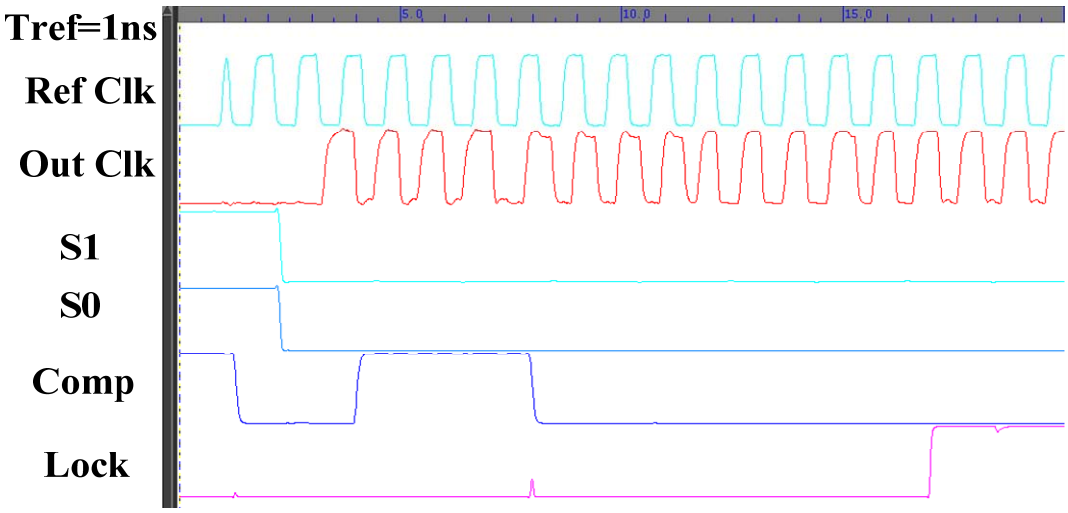


Figure 5.15: The operation of proposed lock-in unit at 1GHz

5.2.4 A Robust, Low Power, And Duty Cycle Immunity Frequency Multiplier

Recently, several DLL-based clock generators have been proposed to overcome the difficulty of frequency multiplication with DLLs and utilize the several inherent advantages of DLLs over PLLs. In [20], the DLL-based local oscillator for PCS applications has been proposed; however, it requires LC-tank circuit, which consumes large power and area; in addition, the frequency multiplication factor cannot be changed once the LC-tank value is chosen. A digital programmable frequency multiplier is proposed in [23] for dynamic frequency scaling application. However, it consists of pulse generator and many AND gates to generate the multiplied clock, which results in large power dissipation and poor jitter; moreover, the pulse generator may suffer from serious malfunction in different voltage domain.

Duty cycle of the frequency multiplied clock is required 50% in many high-performance applications. In [35] [36], the multiplied output clock is not guaranteed 50% when the input duty cycle varies. Therefore, the robust, low power,

and input duty cycle immunity DLL-based frequency multiplier is proposed for multiple frequency generation applications. In addition, it posses programmable ability and can change between 0.5X, 1X, 2X, and 4X at fast time. We will describe each frequency generation in detail as follow:

1. *05X of input clock frequency:* Figure 5.16 shows the 05X frequency generation circuit. It is similar to the divide-by-two circuit with duty cycle immunity.

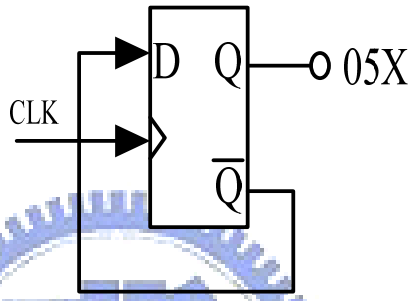


Figure 5.16: 05X frequency generation circuit

2. *1X of input clock frequency:* Figure 5.17 shows the 1X frequency generation circuit. Four phases (Φ1~Φ4) are fed into SR-latch and combine with another SR-latch. Figure 5.18 shows output 1X frequency with 50% duty cycle when input has 25% and 75% duty cycle.

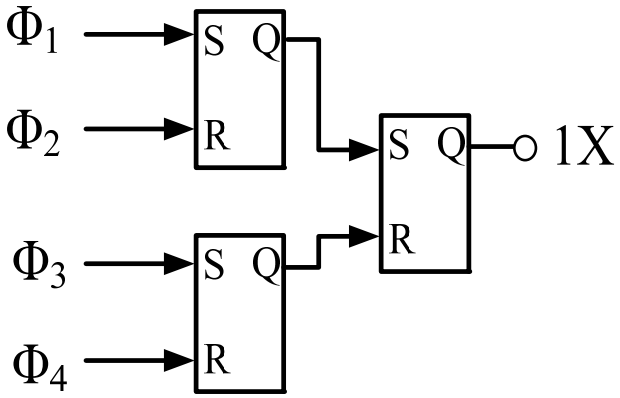
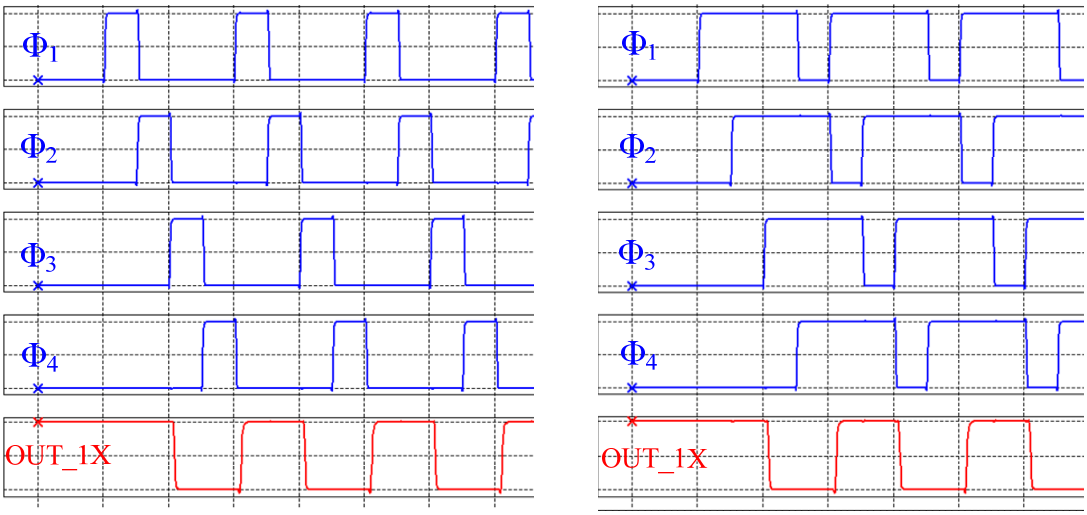


Figure 5.17: 1X frequency generation circuit



(a) (b)

Figure 5.18: Out_1X of input clock frequency when input (a) 25% duty cycle (b) 75% duty cycle

3. *2X of input clock frequency:* Figure 5.19 shows the 2X frequency generation circuit. Four phases (Φ1~Φ4) are fed into SR-latch and combine with a OR gate. Figure 5.20 shows output 2X frequency with 50% duty cycle when input has 25% and 75% duty cycle.

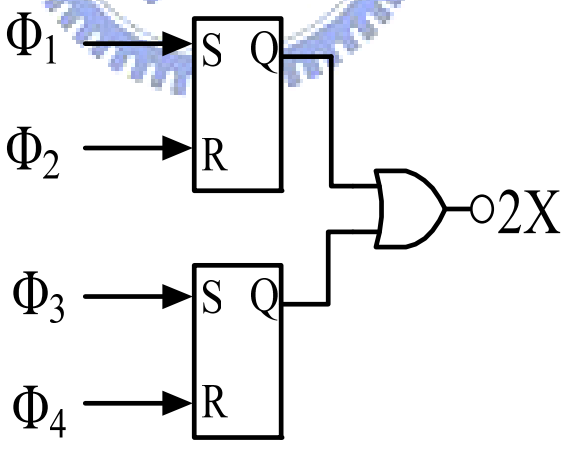
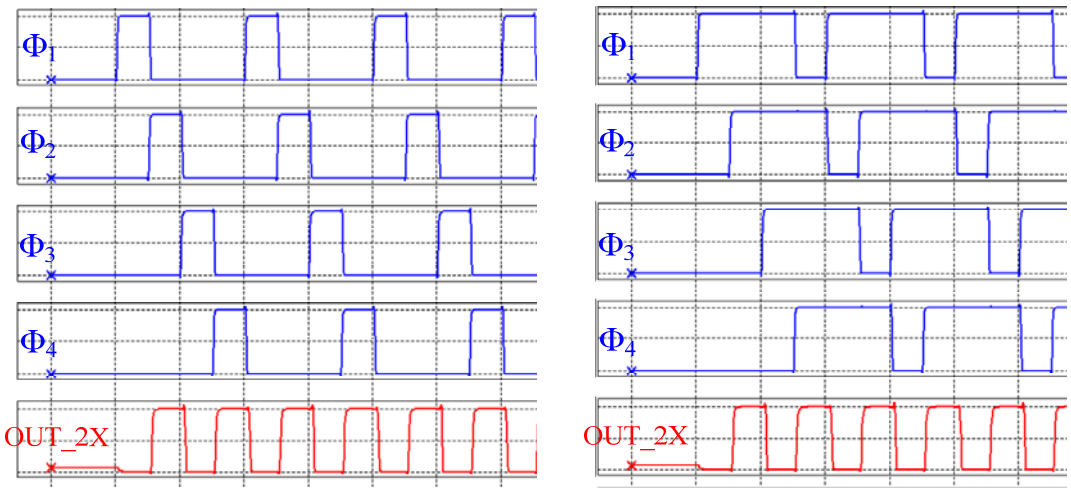


Figure 5.19: 2X frequency generation circuit



(a) (b)

Figure 5.20: Out_2X of input clock frequency when input (a) 25% duty cycle (b) 75% duty cycle

4. *4X of input clock frequency:* In general, DLL-based frequency multiplier require 2N delay cells to generate $F_{out} = F_{ref} \times N$. It not only causes large power consumption but also increase the in-lock phase error. The proposed multiplier only need N+1 delay cells and can generate $F_{out} = F_{ref} \times N$. Thus, the phase error and power could be reduced significantly. Figure 5.21 shows the 4X frequency generation circuit. It is composed of one replica delay cell of main delay line and a XOR gate. We illustrate its operation by an example. When DLL is locked, we assume the binary-weighted control words (BWCW) is 100, and each phase different is Td. Then, the 2X frequency are fed to replica cell, which BWCW is 010, and each phase different is Td/2 in this situation. Combine frequency 2X and 2X_D with the XOR gate, frequency 4X is generated. Figure 5.22 shows timing diagram of 4X frequency generation.

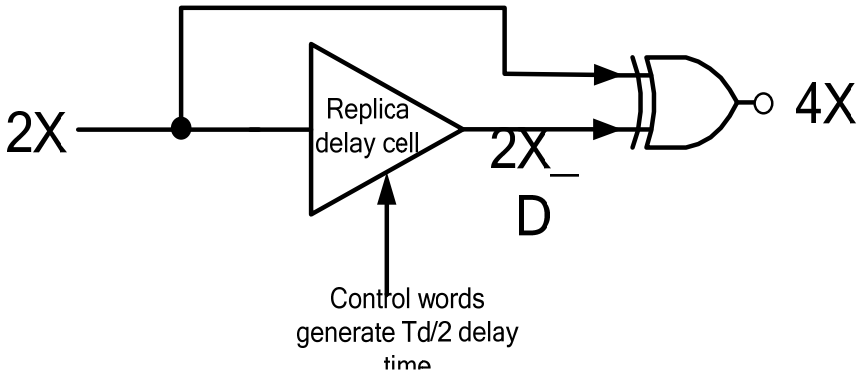


Figure 5.21: 4X frequency generation circuit

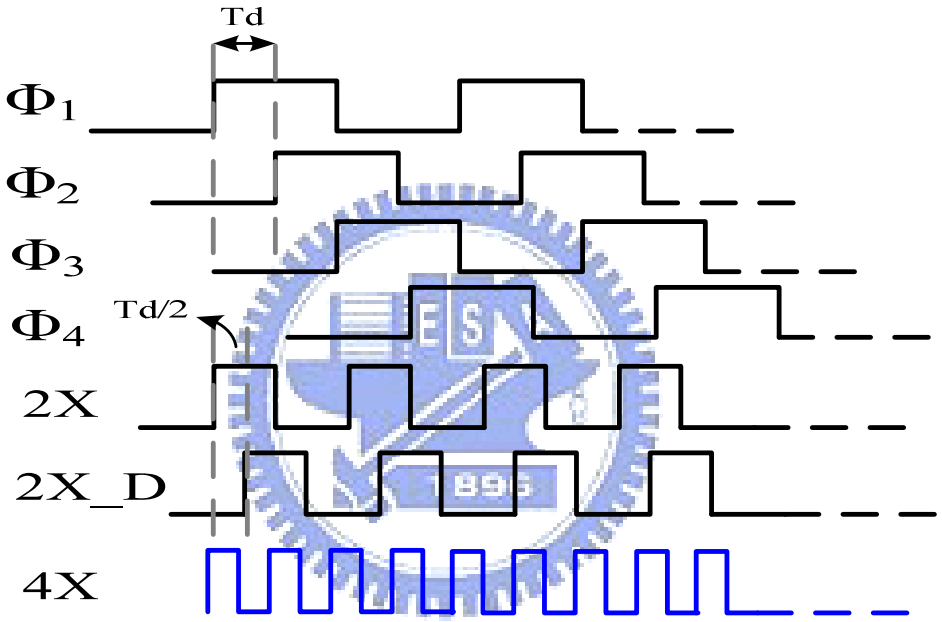


Figure 5.22: timing diagram 4X frequency generation circuit

In conclusion, the proposed frequency multiplier takes advantages of robust, low power, programmable, and duty cycle immunity, which is suitable for dynamic frequency/voltage scaling on low-power microprocessors, portable device, and SoC implantation.

5.2.5 Dynamic Frequency Monitor Circuit

Conventional SAR-type DLLs may suffer from a dead-lock problem due to environmental variations after DLLs are locked. It may require external reset signal to

reset clock. To detect frequency variation automatically after lock, the dynamic frequency monitor (DFM) circuit is proposed.

Figure 5.23 shows the DFM circuit; it operates as an additional phase detector (PD). After the DLL achieves lock, the lock-in unit (LU) enables DFM circuit and receives the lock signal from PD. When the lock signal goes from high to low, the DLL is out of lock. The DFM generates internal pulse to reset the clock. To avoid the pulse disappearing, the pulse width should be properly designed; therefore, the delay Td is added to the DFM reset path. The total compensation procedure is no more than 28 reference clock cycles. Figure 5.24 shows the timing diagram of the DFM circuit.

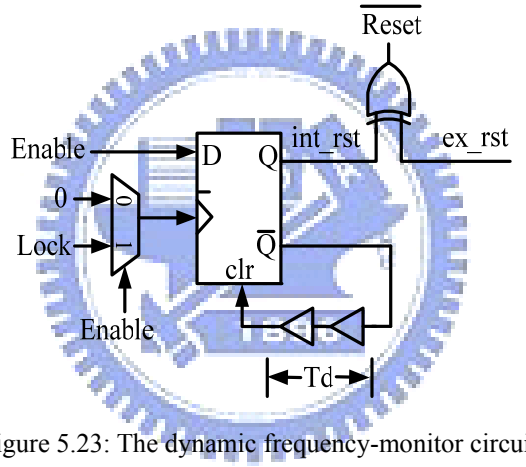


Figure 5.23: The dynamic frequency-monitor circuit

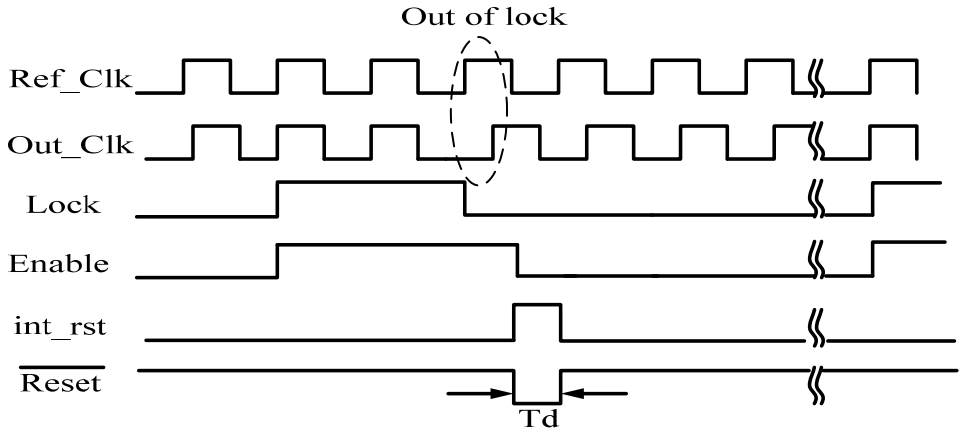


Figure 5.24: Timing diagram of the DFM

5.3 DESIGN IMPLEMENTATION

An all-digital programmable DLL-based frequency multiplier (ADP-DLLFM) is implemented in UMC 90 nm standard CMOS technology with 0.3V-1V supply voltage. The major feature is that the ADP-DLLFM can generate programmable frequency in wide power supply range and wide frequency range without malfunction. The proposed ADP-DLLFM 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 5pF, which is the capacitance of the output pad. Therefore, output buffers are inserted to drive this large loading. The layout view of the ADP-DLLFM and the test chip are shown in Figure 5.25 and Figure 5.26, respectively.

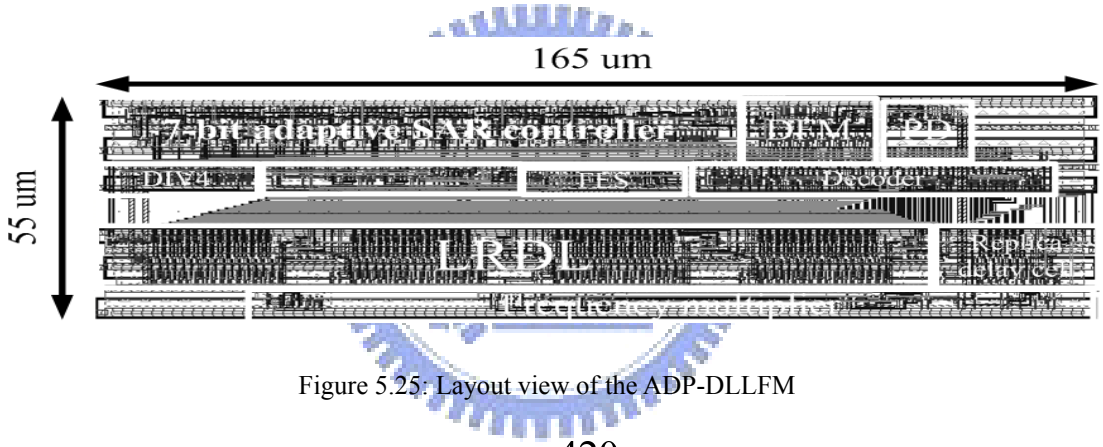


Figure 5.25: Layout view of the ADP-DLLFM

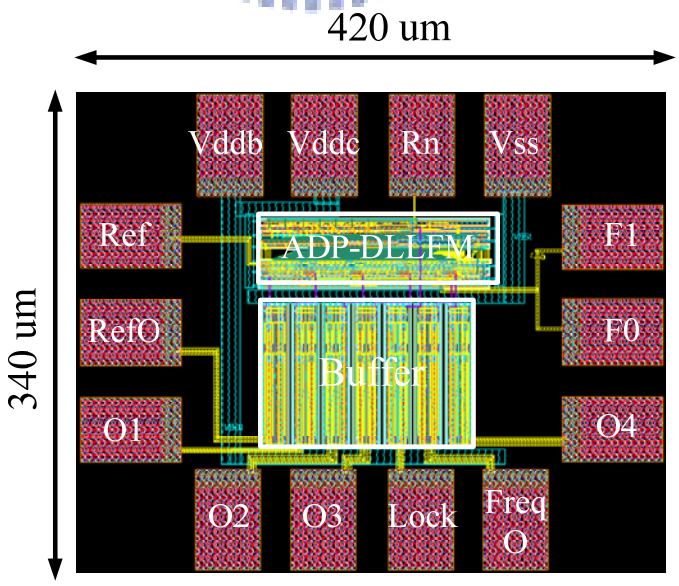


Figure 5.26: Layout view of the test chip

5.4 SIMULATION RESULT

The proposed all-digital programmable DLL-based frequency multiplier (ADP-DLLFM) is implemented in UMC 90 nm standard CMOS technology. The operation frequency range is from 125MHz – 2.5GHz, and available to synthesize (05X, 1X, 2X, 4X) four different frequency with output duty cycle 50%. Based on the proposed adaptive SAR controller, the ADP-DLLFM can operate properly in wide frequency and voltage range; in addition, the locking step is no more than 28 reference clock cycles. The total power consumption is 710 μW at 2GHz with 1.0V power supply, and 4.2 μW at 115MHz with 0.3V power supply. Figure 5.27 and Figure 5.27 and Figure 5.28 show when ADP-DLLFM is locked, it can synthesize four different frequency with power supply 1.0V and 0.3V, respectively.

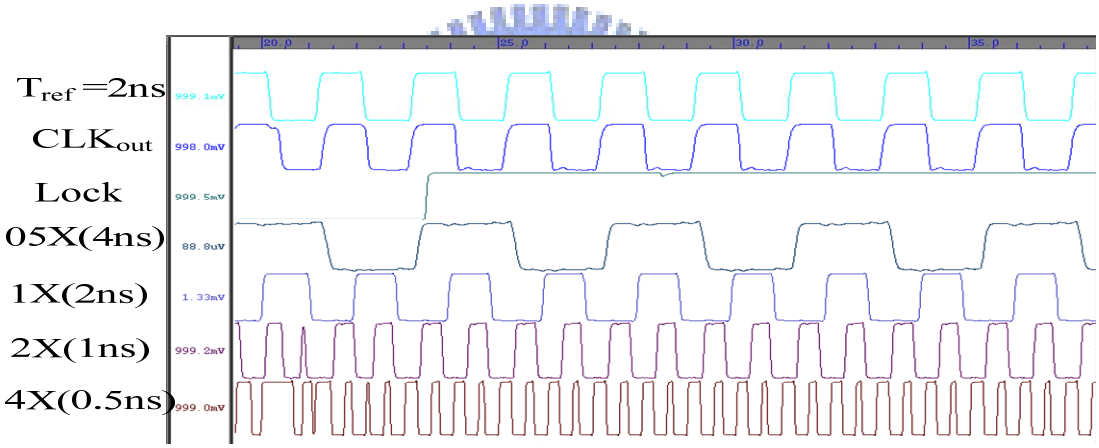


Figure 5.27: Generate four different frequency with 1.0V power supply

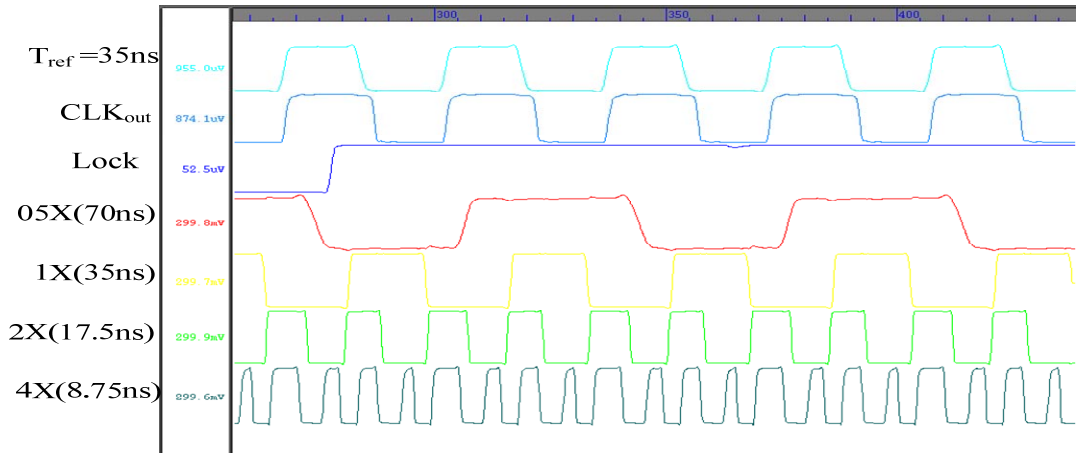


Figure 5.28: Generate four different frequency with 0.3V power supply

The total simulation result of chip implementation summary and a comparison result under the UMC 90nm CMOS technology are shown in Table 3.

Table 3 Comparison among previous works

Design	ISSCC 05[23]	ASSCC 07[39]	ESSCC 08[35]	ICECS 08[38]	TCAS 09[37]	**This work**
Process	0.35um	0.18um	0.09um	0.18um	0.13um	0.09um
Voltage	3.3V	1.8V	1.0V	1.8V	1.2V	1.0V
Output phase	8	8	4	8	8	4
Operating frequency	0.12-1.8 GHz	0.14-3.2 GHz	~2GHz	0.25-2.5 GHz	0.12-2 GHz	0.12-2.5 GHz
Jitter(p-p)	6.6ps @1.3GHz	16.8ps @1.7GHz	9.5ps @2GHz	22.6ps @2.5GHz	19ps @1GHz	22ps @2GHz
Category	Analog	Analog	Digital	Analog	Analog	Digital
50% Duty cycle	Yes	Yes	No	No	Yes	Yes
Input type	Dual	Dual	Single	Single	Dual	Single
Programmable	Yes 0.5x 1x 2x 4x	Yes 0.5x 1x 2x 4x	Yes 1x 2x 4x	Yes 1x 2x 4x	Yes 0.5x 1x 2x 4x	Yes 0.5x 1x 2x 4x
Active area	0.07 mm^2	0.04 mm^2	0.037 mm^2	0.153 mm^2	0.02 mm^2	0.009 mm^2
Power	87mW @1.6GHz	36.7mW @1.7GHz	7mW @2GHz	10.1mW @2.5GHz	21mW @2GHz	20.8mW @2GHz

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

A novel 125MHz-2.5GHz all-digital DLL-based clock generator is implemented in UMC 90nm CMOS technology. The proposed adaptive SAR algorithm achieves fast lock time and extends the locking range with aid of the frequency-estimation selector. In order to accomplish dynamic frequency/voltage scaling application, the programmable frequency multiplier also be proposed. It takes advantages of the robust, input duty cycle immunity, and low power consumption.

The novel leakage-reduced delay unit is proposed to take advantages of mitigating 10% leakage current, insensitive to PVT variations, and not degrading operating frequency of circuit. The simulation results show the proposed DLL-based clock generator exhibits maximum power dissipation 0.71mW when operate in 500MHz, generating 250MHz, 500MHz, 1GHz, and 2GHz four different frequency at the same time. The presented DLL clock generator can be robustly used in embedded memory applications and portable device.

6.2 FUTURE WORK

The DLL-base frequency multiplier combines each multiphase to generate multiple clock frequency. Theoretically, it has better jitter performance than PLL because of no jitter accumulation characteristic. However, due to the delay mismatches or PVT variations, the edges of the multiphase output signals are not equally spaced, and it will induce the fixed pattern jitter at the multiplied clock output. Therefore, a DLL with precise multiphase outputs is necessary. In our future work, we will focus on precise and robust calibration mechanism to compensate the delay mismatch among each delay line.

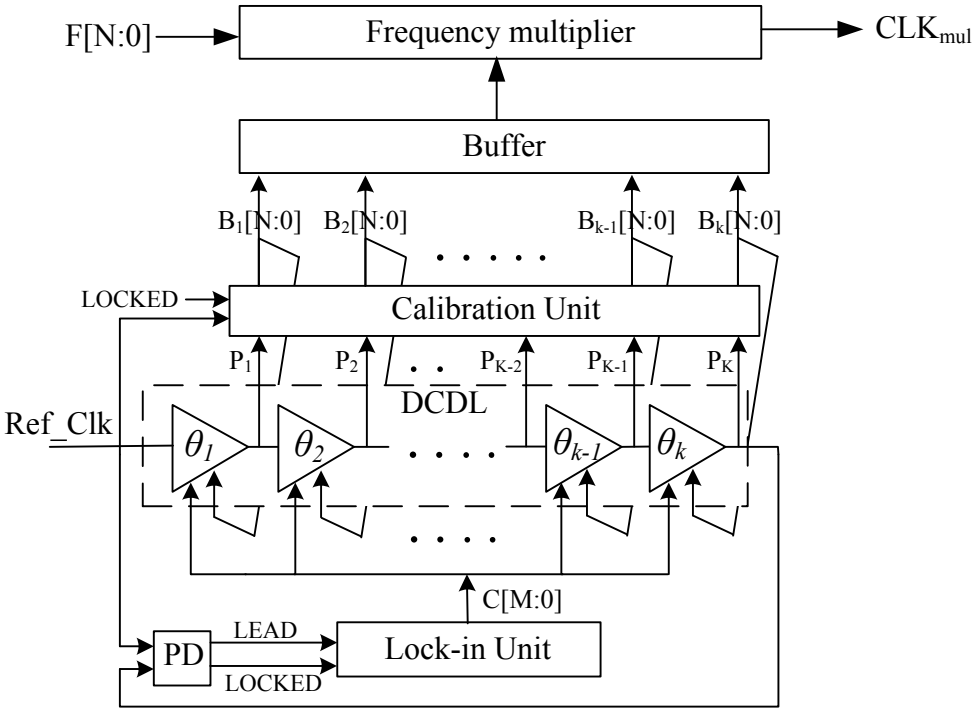


Figure 6.1 ADCGSC clock generator

Figure 6.1 shows the idea of 300mV all-digital DLL-based clock generator with self-calibration (ADCGSC). It includes the proposed clock generator and self-calibration unit. The self-calibration unit reduces the delay mismatch of delay cell, and the clock generator can generate precise and different clock frequency at fast time. As result, the ADCGSC is suitable for portable devices and medical device.

BIBLIOGRAPHY

[1] 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

[2] B. J. Chen “All-Digital Fast-Locking Delay-Locked Loop with Duty Cycle Correction,” M.S. thesis, National Taiwan University, 2006.

[3] F. Herzel et al., “A Study of Oscillator Jitter Due to Supply and Substrate Noise,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II, vol. 49, no. 1, pp.56-62, Jan. 1999.

[4] C. N. Chuang and S. I. 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.

[5] 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.

[6] H. Sutoh, K. Yamakoshi, and M. Ino, “Circuit technique for skew-free clock distribution,” in IEEE Custom Integrated Circuits Conf., 1995, pp. 163–166.

[7] 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.

[8] 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.

[9] 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.

[10] Hamamoto et al., "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.

[11] 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.

[12] 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.

[13] P. L. Chen, C. C. Chung, and C. Y. Lee, "A Portable Digitally Controlled Oscillator Using Novel Varactors," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II, vol. 52, no. 5, pp.233-237, May 2005.

[14] Jinn-Shyan Wang et al.," An Ultra-Low-Power Fast-Lock-in Small-Jitter All-Digital DLL" in IEEE Intl. Solid-State Circuits Conference, Feb. 2005, pp. 422 – 607.

[15] 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.

[16] 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.

[17] Ching-Che Chung, Pao-Lung Chen, and Chen-Yi Lee, "An All-Digital Delay-Locked Loop for DDR SDRAM Controller Applications," VDAT pp.

1-4, April, 2006.

[18] A. Alvandpour, et al., "A 3.5GHz 32mW 150nm multiphase clock generator for high-performance microprocessors," in IEEE International Solid-State Circuits Conference, Feb. 2003, pp. 112–482

[19] G. Chien "Low-Noise Local Oscillator Design Techniques using a DLL-based Frequency Multiplier for Wireless Applications," Ph.D. thesis, University of California at Berkeley, 2000.

[20] 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.

[21] R. Farjadrad, et al., "A Low-Power Multiplying DLL for Low-jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips," IEEE J. Solid-State Circuits, vol. 37, no.12 pp. 1804–1812, Dec. 2002.

[22] 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.

[23] 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.

[24] 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.

[25] H. H. Chang, R. J. Yang, and S. I. Liu, "Low jitter and multirate clock and data recovery circuit using a MSADLL for chip-to-chip interconnection," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I, vol. 51, no. 12, pp.2356-2364, Dec. 2004.

[26] K. Nose et al., "Deterministic Inter-core Synchronization with Periodically All-in-phase Clocking for Low-power Multi-core SoCs," in IEEE

International Solid-State Circuits Conference, Feb. 2005, pp. 296-599.

[27] 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.

[28] Byung-Guk Kim and Lee-Sup Kim, “A 250-MHz–2-GHzWide-Range Delay-Locked Loop,” IEEE J. Solid-State Circuits, vol. 40, no.6 pp. 1310–1321, June. 2005.

[29] Y. Moon et al., “An all-analog multiphase delay-locked loop using a replica delay line for wide-range opearation and low-jitter performance,” IEEE J. Solid-State Circuits, vol. 35, no. 3, pp. 377-384, Mar. 2000.

[30] D. J. Foley et al., “A CMOS DLL-based 2-V 3.2-ps jitter 1-GHz clock synthesizer and temperature-compensated tunable oscillator,” IEEE J. Solid-State Circuits, vol. 36, no.3 pp. 417–423, March. 2001.

[31] B. Fu and P. Ampadu, "Comparative Analysis of Ultra-Low Voltage Flip-Flops for Energy Efficiency," in IEEE Proc. ISCAS, May 2007, pp. 1173-1176.

[32] K. Roy et al., “Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits,” Proceedings of the IEEE, vol. 91, no. 2, pp. 305-327, Feb. 2003.

[33] 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.

[34] 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.

[35] B. Mesgarzadeh *et al.,* “A 2-GHz 7-mW digital DLL-based frequency multiplier in 90-nm CMOS,” 34th European Solid-State Circuits Conference, Sep. 2008, pp. 86-89.

[36] C. H. Lin et al., "A 2.24GHz Wide Range Low Jitter DLL-Based

Frequency Multiplier using PMOS Active Load for Communication Applications," in IEEE Proc. ISCAS, May 2007, pp. 3888-3891.

[37] Koo Jabeom et al., “A Low-Power Programmable DLL-Based Clock Generator With Wide-Range Antiharmonic Lock,” in IEEE TCASII, vol. 56, pp.21-25, Jan. 2009.

[38] K.H Cheng et al., “A wide-range DLL-based clock generator with phase error calibration,” in IEEE ICECS, pp. 798-801, Aug. 2008.

[39] Kyunghoon Chung *et al.,* “An anti-harmonic, programmable DLL-based frequency multiplier for dynamic frequency scaling,” in IEEE ASSCC, 2007, pp. 276–279.