# 國立交通大學

## 電機學院　電子與光電學程

## 碩 士 論 文

在閉迴路上使用資料相位校正器之
10-Gb/s CMOS 時脈與資料回復電路

A 10-Gb/s CMOS Clock and Data Recovery Circuit
with Data-Deskew Buffers in the Closed Loop

指導教授：蘇朝琴　周世傑 教授

研 究 生：楊忠傑

中 華 民 國 九 十 六 年 九 月

# 在閉迴路上使用資料相位校正器之 10-Gb/s CMOS 時脈與資料回復電路

# A 10-Gb/s CMOS Clock and Data Recovery Circuit with Data-Deskew Buffers in the Closed Loop


研 究 生：楊忠傑　　Student : Chungchieh Yang

指導教授：蘇朝琴　教授　　Advisors : Dr. Chauchin Su

周世傑　教授　　Dr. Shyhjye Jou

國 立 交 通 大 學

電機學院　電子與光電學程


碩士論文

A Thesis

Submitted to College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics and Electro-Optical Engineering

September 2007

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 六 年 九 月

# 在閉迴路上使用資料相位校正器之 10-Gb/s CMOS 時脈與資料回復電路

研究生：楊忠傑　　指導教授：蘇朝琴 教授
周世傑 教授

國立交通大學　電機學院　電子與光電學程碩士班

## 摘　　要

本篇論文提出一個適用於晶片上多通道之資料校正(data-deskew)時脈與資料回復(CDR)電路。

此 CDR 系統是藉由調整數位控制延遲線(DCDL)內的資料延遲量來回復單通道 10-Gb/s 之突發資料封包。在最佳選取點獲取之後，資料週期的中點會對準時脈的選取邊緣，同時由通道所造成的偏移也獲得補償。

此 CDR 是一階系統，因此本質上就是穩定的。在傳送和接收端最大時脈抖動差量合於規範的前提下，它可以鎖定的頻率誤差為 1000 ppm。又，閉迴路的特性使得此系統內之抖動為一低通模式。

所有的電路方塊均採用數位電路的實現方式。透過可信度計數器內部所採用之多數決(majority-vote)方式來達到快速鎖定(平均為 110 個位元時間)。在此系統中存在兩個關鍵設計：1)是高速、大擺幅之 CMOS 延遲線設計，2)是滿足迴路延遲之設計條件。

此外，這篇論文使用 TSMC 0.13-$\mu$m CMOS 製程，實現了一個 10-Gb/s 的數位傳接器。

**索引詞彙—高速序列化傳輸鏈，CMOS 傳接器，時脈與資料回復，延遲鎖定迴路，鎖相迴路，數位控制延遲線，相位校正緩衝器，可信度計數器。**

# A 10-Gb/s CMOS Clock and Data Recovery Circuit with Data-Deskew Buffers in the Closed Loop

Student: Chungchieh Yang

Advisors: Dr. Chauchin Su
Dr. Shyhjye Jou

Degree Program of Electrical and Computer Engineering
National Chiao Tung University

## Abstract

This thesis proposes a data-deskew clock and data recovery (CDR) architecture for the on-chip multi-channel timing recovery.

This CDR recovers the 10-Gb/s/ch burst data packet by adjusting the data delay in the digitally controlled delay line (DCDL). After the acquisition of the optimal sampling phase, the midpoint of data period aligns to the sampling clock. The data skew between channels is also compensated.

This CDR is first-order and therefore inherently stable. It can track the specified 1000-ppm frequency error as long as the peak-to-peak clock jitter between the transmitter and the receiver sites is confined to the specification. And, the closed loop characterizes the high-band-limited jitter in this system.

All building blocks adopt digital circuits. Fast acquisition (110-bit time in typical case) is achieved by the majority-vote scheme in the confidence counter. Two critical designs exist in this digital-circuit CDR: 1) high-speed large-swing CMOS DCDL deign, and 2) meeting the loop-latency constraint.

In addition, a digital implementation of the 10-Gb/s transceiver is realized in TSMC 0.13-μm CMOS technology.

**Index Terms—High-speed serial links, CMOS transceivers, clock and data recovery, delay-locked loops, phase-locked loops, digitally-controlled delay lines, deskew buffers, confidence counters.**

# 誌　　謝

由衷感謝 蘇朝琴教授與 周世傑教授兩位老師，在個人遲來的學生生涯中給予指導以及教誨。尤其感謝蘇老師帶我進入高速序列化傳輸的領域，除了授予電路設計的知識，並教我做人處事更圓融。

也感謝交通大學在職專班提供的升學途徑。從前在業界做中學，系統與電路多要靠自己摸索。「能夠在校園內接受正規的訓練，是一件相當幸福的事。」

希望將這份喜悅與我的父母家人分享，感謝您們長久以來的支持與付出。還有感謝我的女友 stella，多年的陪伴與鼓勵。

另外，感謝鴻文在學業上無私的指點與協助；感謝仁乾，在困頓的時候與你談話總能帶來安慰。感謝丸子照料實驗室的大小事，讓我們的學習沒有了後顧之憂。還有謝謝盈杰，諸多生活瑣事向你請益總能得到完滿的答覆。

文末，祝福 918 實驗室的學長、同學與學弟妹們，祝福每個人都能找到屬於自己的一片天空。

楊忠傑

Sep 2007

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

SoC (system-on-chip) has become an industrial trend to provide a solution of both high performance and low power consumption. As CMOS technology advances rapidly, die size of couple centimeters and gate count of more than hundreds of million have been brought to the reality. High-throughput data communication is demanding.

A high-speed serial link transports serialized data stream from the near site to the far site. It is a common way to save the routing cost of the low-speed parallel channels. As the computing/processing capability of a digital system is greatly improved by the advancing technology, the finite bandwidth of a single serial link may no longer meet the required bandwidth. It is necessary to apply *parallelism* back to the serial links. The performance of these serial links turns out to be the performance metric of the entire system.

The proposed clock and data recovery (CDR) system is for the on-chip

burst-mode timing recovery, and is implemented with digital circuits. It adjusts the delay of the data by deskew buffers and aligns the midpoint of a data period to the clock so as to recover the data synchronously. It is called a data-deskew CDR, or simply Deskew CDR.

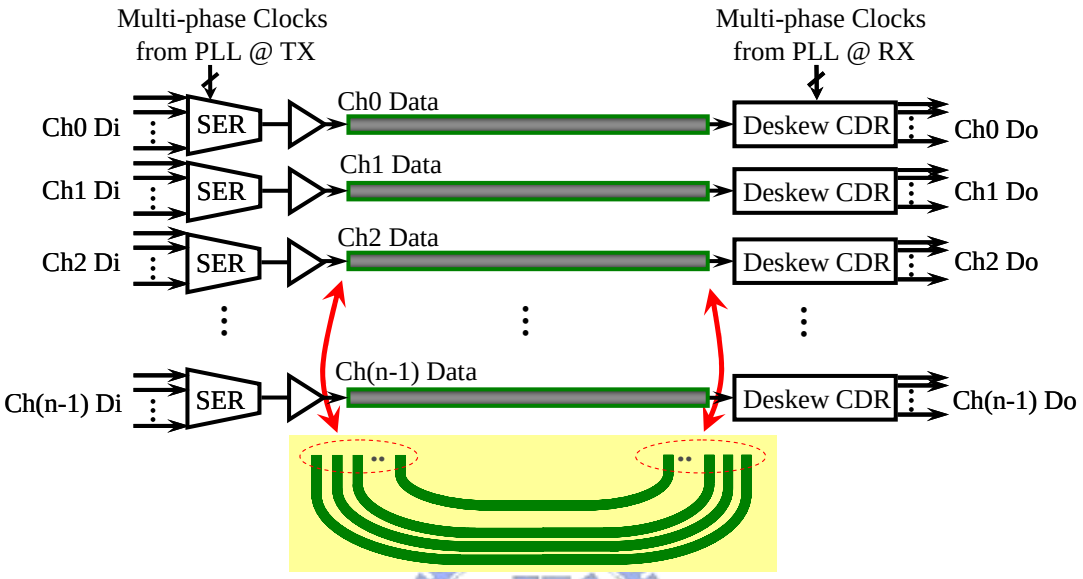



Fig. 1.1: Multi-channel timing recovery by multiple Deskew CDRs

Multiple Deskew CDRs target on multi-channel timing recovery. Fig. 1.1 shows n-channel timing recovery by n sets of Deskew CDRs. Data skews between channels consist of static and dynamic components. The illustrated channel routing mismatch refers to the static aspect. And, driving source mismatch, substrate noise under channels, electromagnetic interference during transmission, etc refer to the dynamic aspect. Deskew CDRs compensate the total data skews between channels, and accomplish the timing recovery.

Fig. 1.1 also shows the environment setup for multiple links. A single PLL located at the transmitter site (TX) provides multi-phase clocks to serializers. Another PLL located at the receiver site (RX) provides multi-phase clocks to the CDRs. The clock jitter between PLL at TX and PLL at RX is specified, and it refers to the tuning range of digitally controlled delay line (DCDL) directly.

These two PLLs introduce a relatively simple environment for multi-channel timing recovery, as compared to that of conventional PLL-based CDRs.

## 1.2 Features

The 10-Gb/s Deskew CDR is a first-order recovering system, and therefore it is inherently stable. High-frequency jitter is filtered because of the loop filter. There is no jitter accumulation due to the absence of $(1/s)$ from an oscillator inside the loop. A first-order system intends to track the phase error. Still, it can track the frequency error between TX and RX to some extent.

All the building blocks adopt digital circuits for design robustness, since digital circuits benefit more from the scale-down technology than most analog circuits. Besides, analog circuits, such as the conventional charge pump and the passive R-C filter, still suffer from the matching issue. In this work, a digital confidence counter serves as the loop filter to the CDR. Fast acquisition is achieved by the majority vote scheme, which is implemented as a comparator inside the confidence counter.

Deskew CDR focuses on the design of DCDL. The high-speed large-swing CMOS delay line provides the monotonic delay adjustment. A tuning range of more than 1.4 UI is achieved in this work. Static CMOS inverters compose the main delay stages for the delay line. Large noise margin is guaranteed by the large-swing behavior and also by the hysteresis characteristic of the delay cell.

A wider tuning range can be achieved under a lower supply as long as the DCDL-induced jitter meets the specification. As the supply gets lower, the crossing point of the delayed data is self-adjusted due to the balanced pull-up/pull-down CMOS inverter.

Besides, an implementation of the 10-Gb/s transceiver in TSMC 0.13-μm CMOS technology is demonstrated in this work.

## 1.3 Organization

This dissertation comprises seven chapters. The motivation and features of the CDR are described in this chapter.

Chapter 2 gives an overview of the world's CDR architectures, which is far beyond the PLL-based and the oversampling architectures. It describes the operation principles, the issues, and discovers the possibilities on various architectures.

Chapter 3 analyzes the system-level behaviors of the CDR. Topics on system specifications, design parameters, and simulations are involved. Discussions and simulations on various cases, such as *noise profiles, the frequency tolerance, the loop latency, and the frequency error*, can be found in this section.

Chapter 4 depicts the circuit-level implementation. *The high-speed large-swing DCDL design* and *meeting the loop-latency constraint* are described. To minimize the loop latency, the pseudo-NMOS scheme is adopted. Demonstrated can be found in the carry-look-ahead adder of the comparator, or the TFF up-down counters of the confidence counter and the FSM.

Chapter 5 shows the digital implementation of the 10-Gb/s transceiver. It describes the considerations for design and test, as well as different operation modes. The CDR is verified in nominal mode and/or debug mode, while the phase resolution of DCDL is measured in bypass mode. The multi-phase clock generator, the serializer, and the 10-Gb/s output buffer are also discussed in this section.

In the high-speed domain, the circuit layout is critical and dominates the final eye open. Chapter 6 shows the chip layout, grid design, and describes the layout guidelines for high-speed circuits. The guideline of *source coupling* is especially emphasized. Post simulations of DCDL, the CDR, and the full-chip transceiver are given in this section.

The final section, Chapter 7, shows the specification table of Deskew CDR. It compares the power/area among several CDR systems.

# Chapter 2

# Overview of the World's CDR Architectures

Different applications require different CDR systems to the world. The types of CDR systems reflect on the modes of recovering process: continuous vs. burst, closed loop vs. open loop, filter-based vs. oversampling, clock delay vs. data delay, and digital vs. analog, and etc.

For conventional CDR systems, we have PLL-based and oversampling CDR architectures. They are well-explored and have their own traditions. But there are more candidates for applications of the timing recovery. Following the classification in [1][1], we categorize CDR systems into *1) PLL-Based, 2) Blind Oversampling, 3) DLL-Based, 4) Gated VCO, and 5) Alternative & Hybrid* architectures.

In this section, some of the demonstrated systems are history, while some of them are state-of-the-art. This section mainly focuses on the world-view, the variety, and the possibilities of CDR systems.

[1] A presentation document introduces the world's CDR systems on the internet by Çobanoğlu in 2006. The original classification is 1) PLL-Based, 2) Delay-the-Data, 3) Gated VCO, 4) (Semi-)Blind Oversampling, and 5) FSM-Based.

# 2.1 PLL-Based CDR

PLL-based CDR systems are suitable for continuous mode operation. They can be characterized as single loop [3]-[5] and dual loop [6]-[9]. In general, a PLL-based CDR system refers to an Nth-order system, where $N \geq 2$. It is usually implemented with analog circuits due to the inherently continuous characteristics.

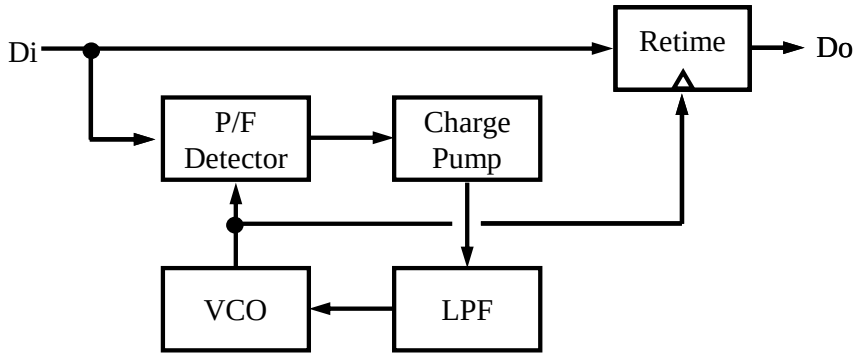



Fig. 2.1: A generic PLL-based CDR with a charge pump

Fig. 2.1 shows a generic PLL-based CDR architecture. It consists of phase and/or frequency detectors, a charge pump, a loop filter, and a VCO. For the dual-loop architecture, the entire recovery process includes the slow pull-in process by frequency detectors and then the lock-in process by phase detectors in sequence.

Back to 1985, a PLL-based system [3] was proposed for clock and data extractions from NRZ data. It employs an active SAW filter in the loop for the band-pass filtering instead of the architecture with a charge pump and a passive filter. After the charge pump becomes popular, a second-order low-pass filter of 'C // R-C' structure is also welcome [5], [8]. The second-order filter composes a third-order system, so that phase step, frequency step, as well as the accelerative frequency variation can be tracked.

A high-order PLL-based system is well known as its high performance. However, it doesn't suit burst-mode applications because of 1) the slow pull-in process, and 2) clock drifting at the case of no input.

Besides, there exists jitter peaking phenomenon [2] in the high-order system. To take the simplest case for instance, consider a second-order system. The closed-loop transfer function is expressed in (2.1).

$$H(s) = \frac{2\varsigma w_n s + w_n^2}{s^2 + 2\varsigma w_n s + w_n^2}, \approx \frac{2\varsigma w_n}{s + 2\varsigma w_n}. \quad (2.1)$$

The approximation of (2.1) is made by assuming that damping factor $\varsigma$ is large (such as 10) and $w_n^2$ is small enough and can be neglected. The approximated loop bandwidth is then derived in (2.2). And the corresponding zero and poles are given in (2.3)-(2.5).

$$w_{-3dB} = 2\varsigma w_n. \quad (2.2)$$

$$w_Z = -\frac{w_n}{2\varsigma}. \quad (2.3)$$

$$w_{P1} \approx -\frac{w_n}{2\varsigma} - \frac{w_n}{8\varsigma^3}. \quad (2.4)$$

$$w_{P2} \approx -2\varsigma w_n + \frac{w_n}{2\varsigma} + \frac{-w_n}{8\varsigma^3}. \quad (2.5)$$

From (2.3) and (2.4), the first pole locates behind the zero in absolute value. The jitter peaking phenomenon is then introduced in the closed-loop transfer plot as shown in Fig. 2.2.

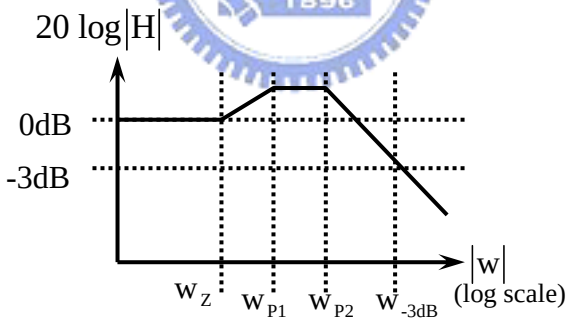



Fig. 2.2: Jitter peaking phenomenon

The jitter peaking $J_P$ is

$$J_P = w_{P1}/w_Z \approx 1 + 1/4\varsigma^2. \quad (2.6)$$

To express (2.6) in dB, we get

$$J_P \approx 8.686 \ln(1+1/4\varsigma^2) \text{ dB}. \quad (2.7)$$

The amount of jitter peaking in (2.7) can be eliminated by over-damping the loop; that is applying large $\varsigma$. But it results in slow response of the lock acquisition.

■ **Example: Savoj2001** [4]

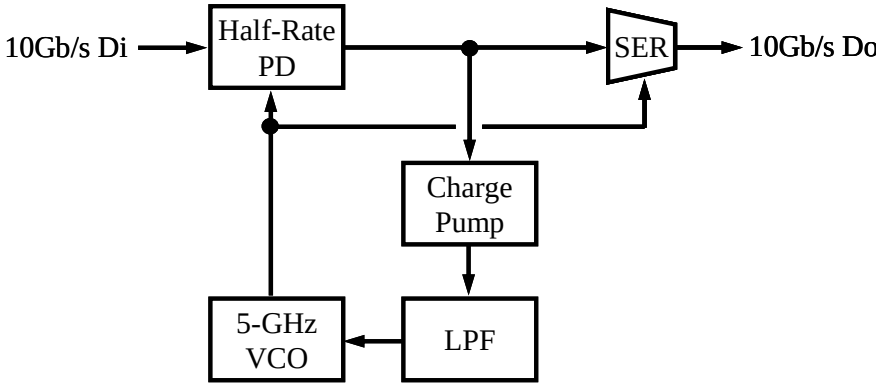



Fig. 2.3: The PLL-based CDR, Savoj2001

■ **Example: Savoj2003 [6]**

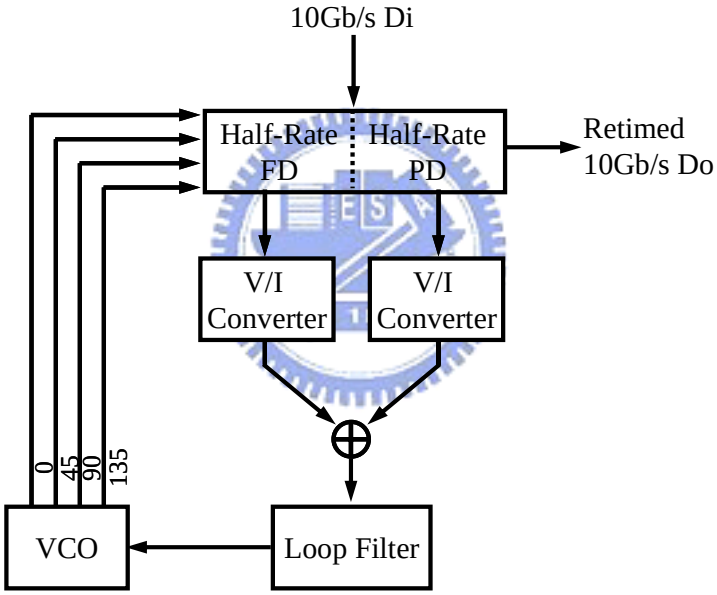



Fig. 2.4: The PLL-based CDR, Savoj2003

# 2.2 Blind Oversampling CDR

A blind oversampling architecture, shown in Fig. 2.5, is implemented with digital circuits, and can handle both continuous and burst-mode timing recovery. It oversamples the data and chooses the optimal clock phase according to the extracted edges information in decision circuit. The decision scheme can be either majority-voting [10] or center-picking [11], while the previous is less superior. [12]

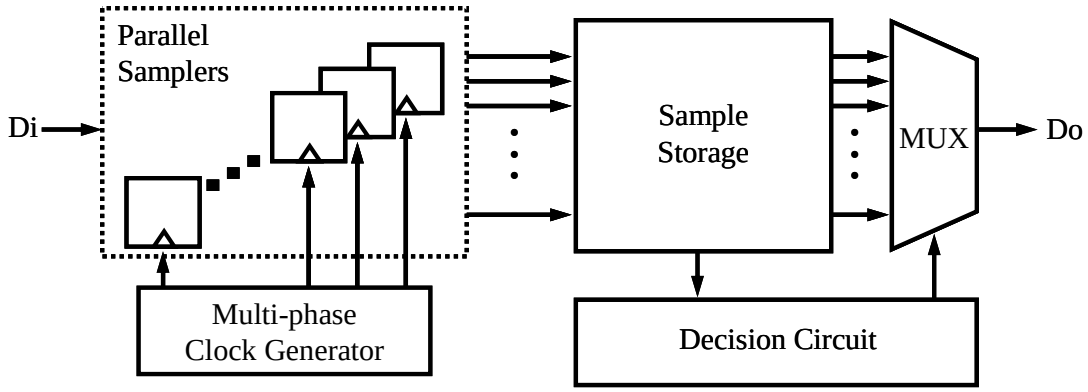



Fig. 2.5: A generic oversampling CDR, Kim&Jeong2003 [13]

A blind oversampling CDR tracks the high-frequency jitter of input data stream well, while the limited size of storage causes a limitation on tracking the low-frequency jitter.

Different from most CDR systems, this architecture eliminates the need on the acquisition time but requires extra hardware for executing algorithm and introduces processing latency to the data recovery.

The phase picking scheme accompanies static offset error on each sampling, because neither the data nor the clock phases are adjusted. The maximum offset error is $(0.5\ \mathrm{UI} / \mathrm{OSR})$, where OSR denotes the oversampling ratio. Although this offset error can be suppressed by raising the oversampling rate, but in practical cases it encounters issues like: 1) A high OSR implies high-accuracy phase resolution for each sampling, which is always a challenge. 2) The input capacitance of phase detectors grows with OSR. That is especially critical to high-speed application. In the conventional way, 3x-oversampling is widely-used.

■ **Example: C.K.Yang98** [14]

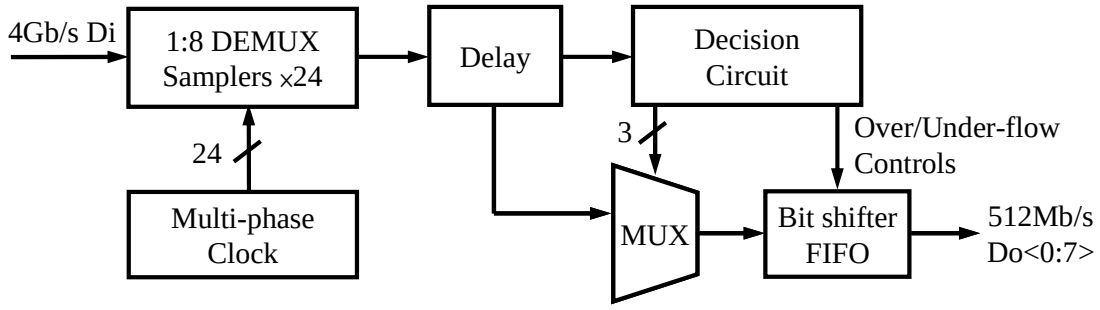



Fig. 2.6: The blind oversampling CDR, C.K.Yang98

In Fig. 2.6, the sample storage is denoted as a delay block, and the decision circuit controls the multiplexer as well as the FIFO at the last stage. The FIFO is implemented with an 8-bit shifter. It handles both the overflow and underflow cases when the phase error, which is mainly caused by the frequency error, accumulates more than 1-bit time.

# 2.3 DLL-Based CDR

A DLL-based CDR can be regarded as a simplified version of PLL-based architecture. It is a closed-loop first-order system without jitter peaking phenomenon. In this system, only the phase delay is a variable. Implementations of DLL-based CDR can be either analog or digital, while the latter is the major trend in recent days.

According to the subject of delay adjustment, it can be distinguished as 1) clock-interpolation, and 2) data-deskew architectures. The clock-interpolation architecture can handle continuous timing recovery by the phase-rotation scheme, but this phase rotation needs additional hardware, such as the FIFO stage of oversampling architecture in Fig. 2.6, to handle the overflow/underflow condition.

As for the data-deskew architecture, it is a straight concept to adjust data instead of clock. It introduces a simple synchronization behavior by the shared and untouched global clock. But it is mainly limited by the data tuning range, and therefore is only suitable for burst-mode applications.

## 2.3.1 Clock-Interpolation CDR

Fig. 2.7 shows an example of clock-interpolation CDR by E. Lee. The 8 clock phases are adjusted by the interpolation scheme, which is generated from the phase controller, and finally the sampling clock phases align to the midpoint of data duration. The *receive amplifiers block* consists of amplifiers and phase detectors. Here the full-rate data is de-multiplexed into 4 quarter-rate data inherently. Even though digital circuits implement the logic function in the phase controller block, the entire CDR implementation also adopts analog circuits.

Fig. 2.8 shows a clock-interpolation CDR for multi-channel timing recovery by Kreienkamp. It adopts analog circuitry to achieve high speed and fine phase resolution. Differential charge pump and two capacitors contribute the single pole to

the system. The phase interpolator is the conventional analog current-steering scheme, and just like those PLL-based CDR systems, the phase resolution is limited by the discrete steps, which is introduced by charge pump. The chip is fabricated in 0.11-μm CMOS technology, and its power consumption is 220-mW at a supply of 1.5 Volt.

But for continuous recovery, it lacks of description about phase-rotation of these CDR macro-cells.

- **Example: E.Lee2001** [15]

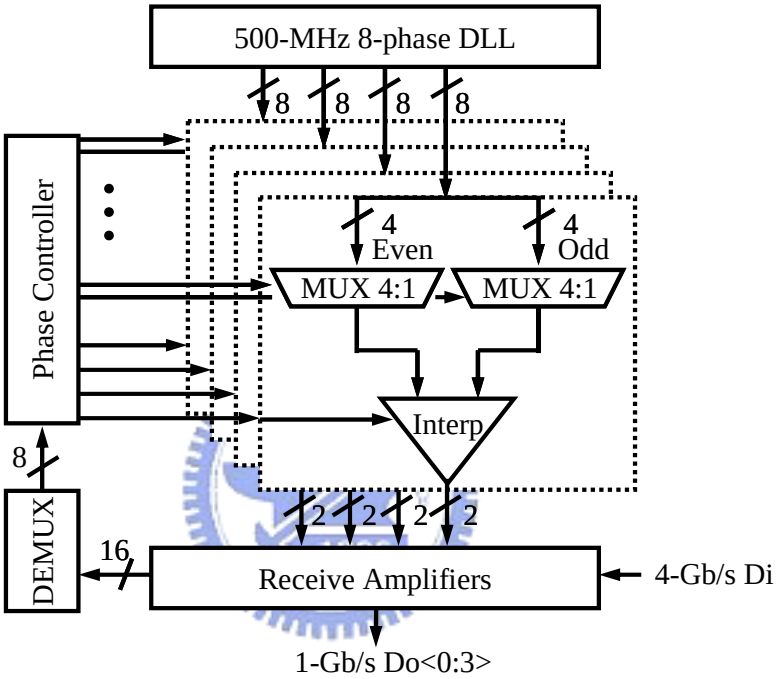
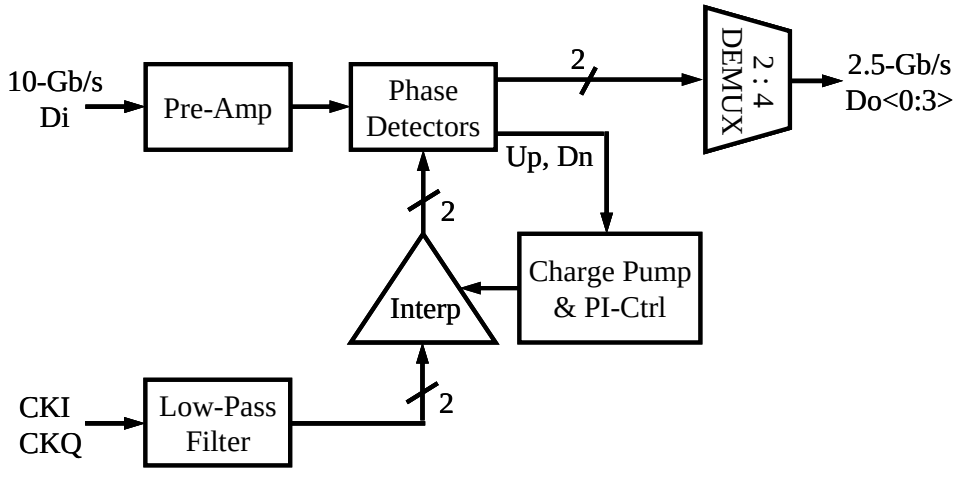
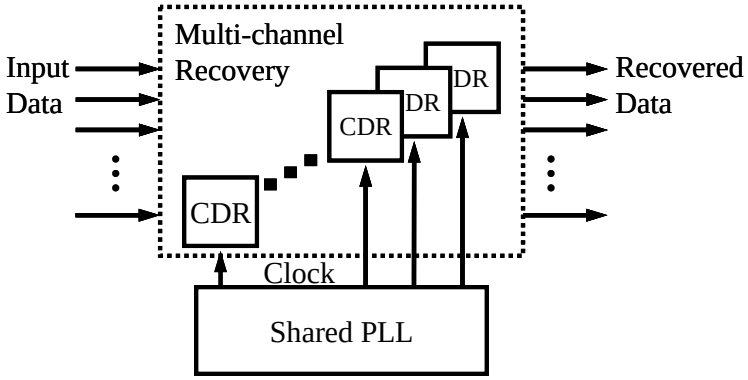



Fig. 2.7: The clock-interpolation CDR, E.Lee2001

- **Example: Kreienkamp2005** [16]




**(a)**




**(b)**

Fig. 2.8: The clock-interpolation CDR, Kreienkamp2005
(a) the CDR, (b) the multi-channel configuration

### 2.3.2 Data-Deskew CDR

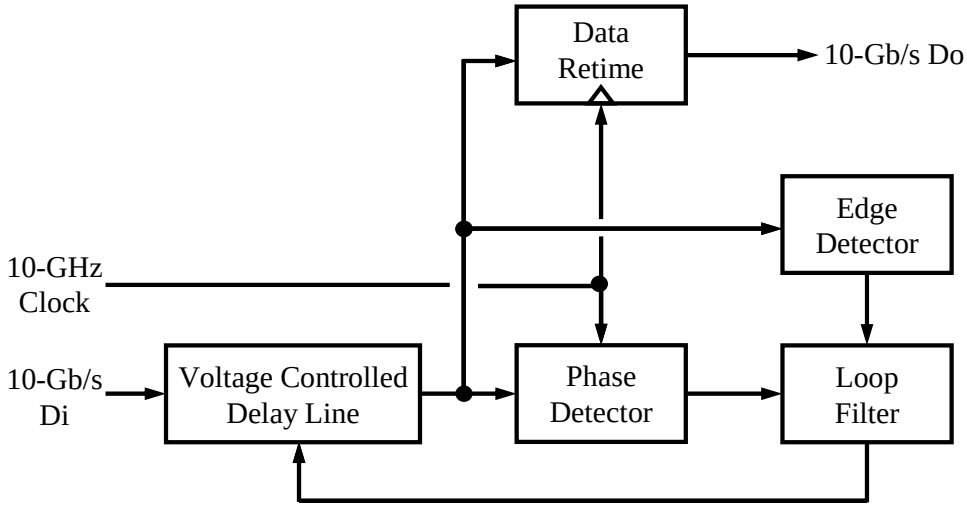
Fig. 2.9(a) shows the 10-Gb/s data-deskew CDR for multi-channel burst-mode applications proposed by Wong. It is a full-rate analog implementation, and fabricated in both AlGaAs/GaAs and InGaP/GaAs HBT technology, where $f_t \sim 50\,\text{GHz}$, $f_{max} \sim 60\,\text{GHz}$ and $\beta \sim 40$. The voltage controlled delay line, phase detector, and loop filter compose the delay lock loop. In addition, it employs an edge detector circuit to adjust the time constant of the loop filter. Fig. 2.9(b) shows the phase detector circuit. The detector's output is generated from the transition edge of input and its asynchronous delay.

The achieved tuning range is 2 UI or 200 ps. It claims to be capable of a 12.5-kbit data packet but under the assumption that frequency error for all clocks is within 20 ppm. The 20-ppm error is far less than the conventional estimation of 200 ppm.

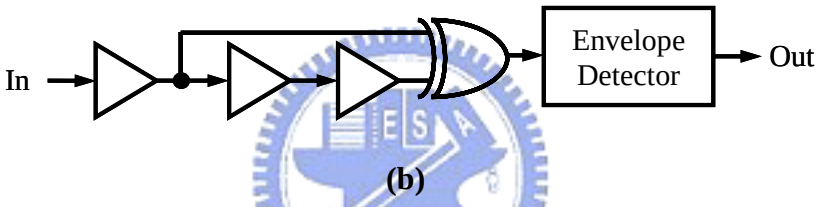
Fig. 2.10 shows a digital implementation of data-deskew CDR by Lu. The confidence counter replaces the conventional loop filter. The cascaded delay cells compose the DCDL block. Coarse and fine tune functions are available. The coarse function is implemented by the on/off state of tri-state buffers in the chain, and the fine function is implemented by the added amount of capacitive load.

It is fabricated in 0.18-μm CMOS technology, and the achieved tuning range is 1 UI, or 400 ps, for the 2.5-Gb/s operation. Due to the insufficient tuning range, this implementation is not going to handle any frequency error.

■ **Example: Wong96** [17]




**(a)**




**(b)**

Fig. 2.9: The data-deskew CDR, Wong96 (a) Architecture (b) Edge Detector

■ **Example: Lu2005** [18]

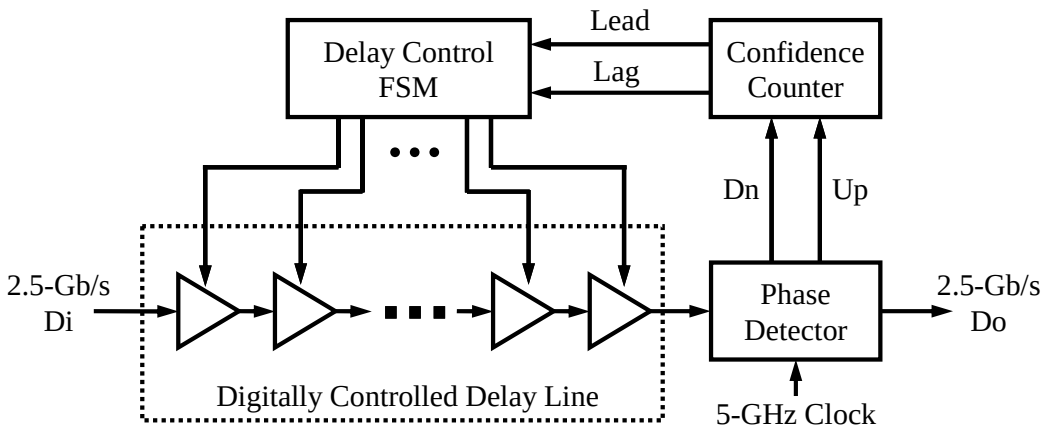



Fig. 2.10: The data-deskew CDR, Lu2005

# 2.4 Gated-VCO CDR

- **Example: Nakamura96** [19]

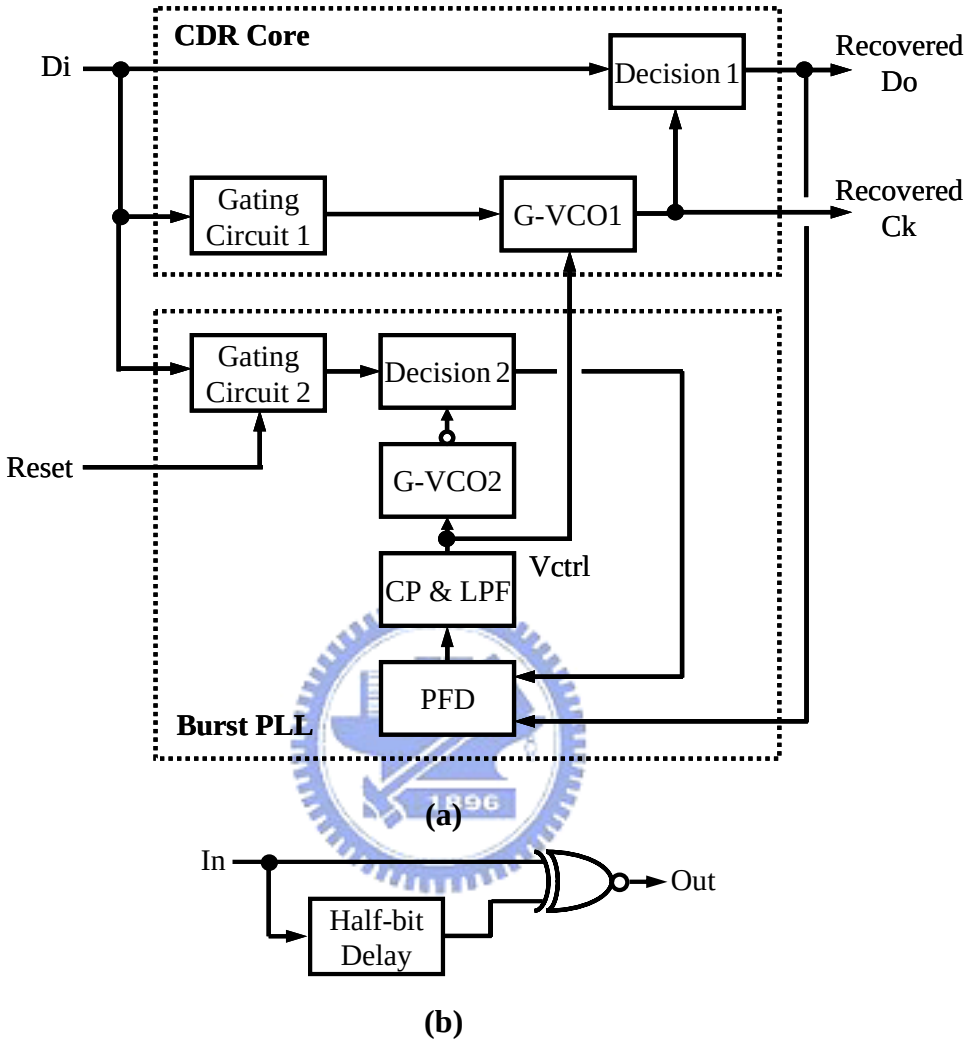



Fig. 2.11: Gated-VCO CDR, Nakamura96 (a) architecture (b) gating circuit

A gated-VCO CDR system was first introduced by Nakamura in 1996. It can fast response to the asynchronous burst input data. In Fig. 2.11(a), the CDR core consists of a gating circuit, a gated VCO, and a DFF at the final stage for retiming the data. This DFF is denoted as *Decision 1* block.

The gating circuit in Fig. 2.11(b) adopts the same scheme as that in Fig. 2.9(b). It detects the transition edge of input data. Consider the gating signal is logic 0, and the Vctrl signal is ready; the gated VCO oscillates by default and is ready to re-initiate an oscillation. As the gating signal validates, the gated VCO re-generates the gated clock instantaneously. In other words, the gating signal re-synchronizes the gated clock,

every time the data transition validates.

This prototype of gated-VCO architecture cooperates with a burst PLL, which provides the control voltage to the CDR. An additional reset action is required after each burst data recovery.

■ **Example: Nogawa2005** [20]

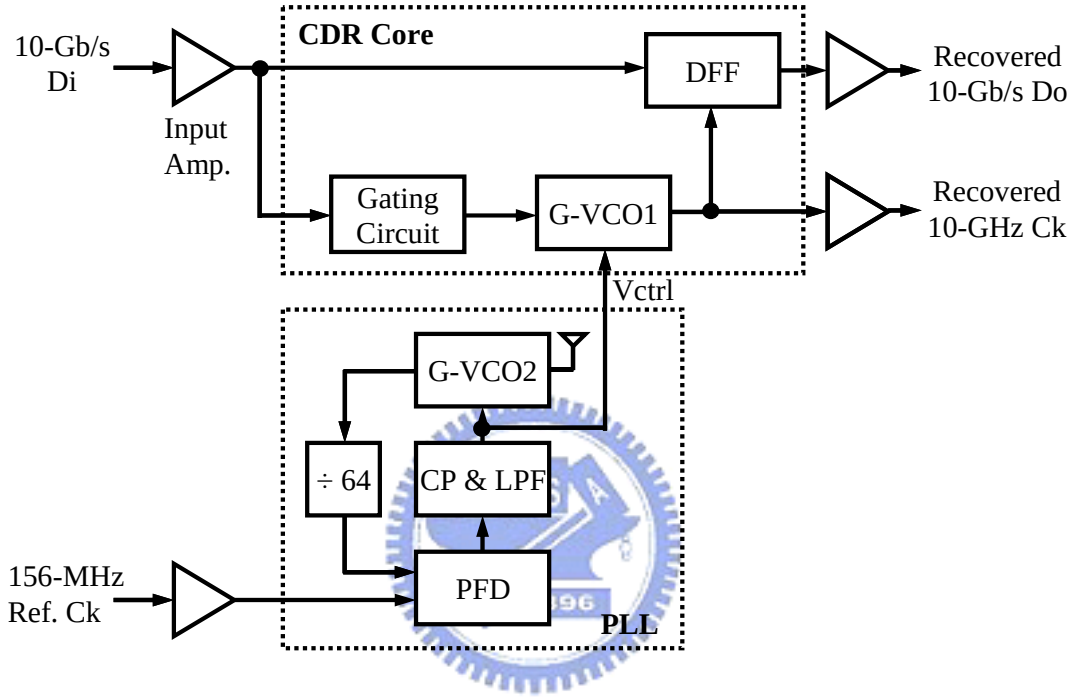



Fig. 2.12: Gated-VCO CDR, Nogawa2005

The implementation in Fig. 2.12 demonstrates a high-performance gated-VCO CDR. It is fabricated in 0.13-μm CMOS technology with the overall area of $2.5\times2.5\ \text{mm}^2$ and power consumption of 1.2 W at a 2.5-V supply. It operates at 10-Gb/s, and is able to extract the recovered clock within 5-bit time.

A new invention of this design is the input amplifier, which applies AC couple and edge detection schemes to accomplish the final comparison in a hysteresis comparator.

Previously in Nakamura's prototype, it employs a burst PLL. But in the later years, a PLL with input reference clock becomes popular for the generation of Vctrl. The gated VCO2 follows reference clock instead of input data. The need for the additional reset action is thus eliminated.

■ **Example: Kaeriyama2003** [21]

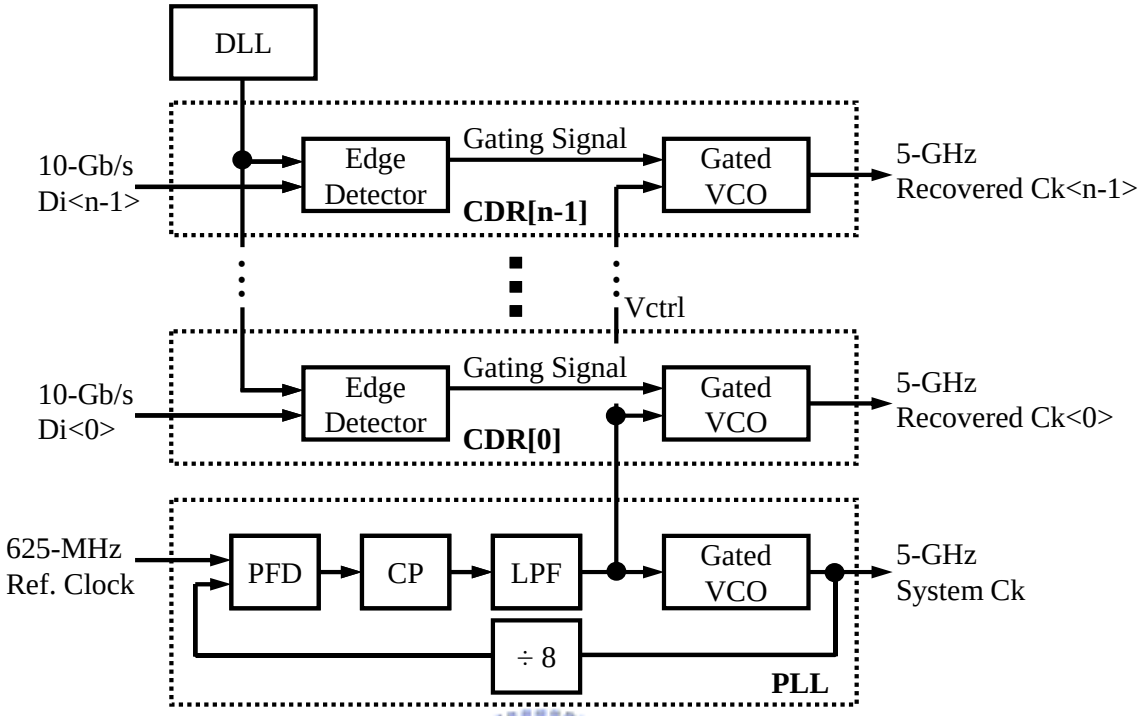



Fig. 2.13: Gated-VCO CDR, Kaeriyama2003

Fig. 2.13 shows the configuration of gated VCO CDR for multi-channel timing recovery. It is implemented in an economic way. First is that gated VCO is inherently low-hardware overhead with the shared control voltage, and second is that all gated VCO operate at half rate.

The CDR macrocell consists of 1) edge detector, 2) a gated VCO, 3) phase detector, and 4) reference voltage generator, where 3) and 4) are not shown in the figure.

The implementation is fabricated in 0.15-μm CMOS technology. Each CDR macrocell recovers 10-Gb/s data with a power dissipation of 50 mW at a 1.5-V supply, while area is $120\times130\ \mu m^2$. But the mentioned area excludes the hardware corresponding to data recovery such as the de-multiplexer for the half-rate data and the retiming circuit.

# 2.5 Alternative & Hybrid

This section introduces alternative CDR architectures, which involve a new recovering method, called FSM-based, and two hybrid architectures.

### 2.5.1 Alternative CDR

■ **FSM-Based, Analui2005** [22]

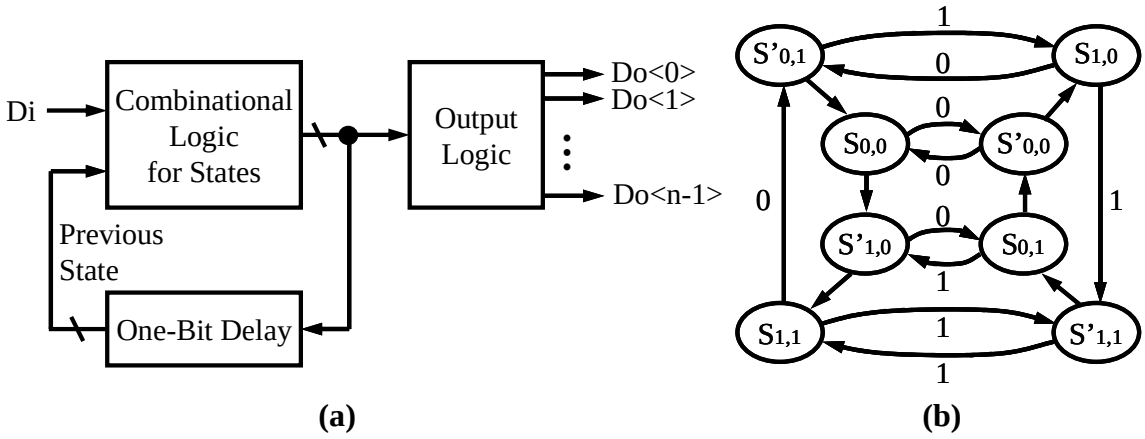



Fig. 2.14: FSM-based CDR, Analui2005 (a) Architecture (b) State Diagram at n=2

The FSM-based architecture is clockless and digital. Fig. 2.14(a) shows the CDR architecture with 1-to-n de-multiplexing, which includes two combinational logic circuits and the one-bit delay circuit. The one-bit delay is implemented with L-C delay cells. The recovered data output depends on the current input and the previous state from the delay line. It is therefore an asynchronous system but synchronized to every transition of incoming data.

The 1-to-n de-multiplexing relaxes the operation rate. Since the state information is kept in the memory of FSM and lasting for n-bit time. This system behaves like open-loop and operates without jitter rejection. The 1-to-n de-multiplexing behavior inherently introduces (1/n) of input jitter to the output.

The implementation operates at 7.5 Gb/s and is fabricated in SiGe technology. It is built with 1-to-2 de-multiplexing. From the data rate and technology, the digital-circuit approach still encounters speed limitation in timing recovery.

### 2.5.2 Hybrid CDR

A hybrid version of oversampling/PLL architecture, called *semi-blind*, is proposed by Ierssel in 2006. Fig. 2.15 shows the architecture. The main system is a blind oversampling architecture, while the second feedback loop shown in the bottom of the figure simulates the PLL-based system. The second feedback loop is composed of a DAC and a loop filter. The original blind oversampling architecture tracks the

high-frequency jitter while the second loop tracks the low-frequency jitter. The jitter tolerance specification at low frequency is greatly (32×) improved by this hybrid version.

Fig. 2.16 shows a hybrid DLL/PLL CDR architecture by T. Lee. The data-deskew path forms the DLL, and the second loop in dashed line refers to the PLL. The system can be either a simple DLL-based CDR by removing the voltage controlled crystal oscillator (VCXO) path or a hybrid DLL/PLL system.

Both DLL and hybrid DLL/PLL architectures provide jitter-peaking-free timing recovery since no zero exists. In summary, DLL loop determines the acquisition speed while the filtering of low-frequency jitter benefits from the PLL loop.

The possibility of the hybrid DLL/PLL architecture can be further explored. Fig. 2.17 [2] shows the weighted control of DLL and PLL by the interpolator. The original design in [26] uses a multiplexer to determine how the loop of the delay line is configured, open vs. closed. When the loop is closed, the delay cells forms an oscillator. In Fig. 2.17, the multiplexer is replaced by an interpolator, and through the weighted control, the behavior can be partial DLL and partial PLL. For instance, the hybrid ratio of DLL to PLL can be 50%-50%, 20%-80%, or anything else.

- **Semi-blind Oversampling CDR, Ierssel2006** [23]

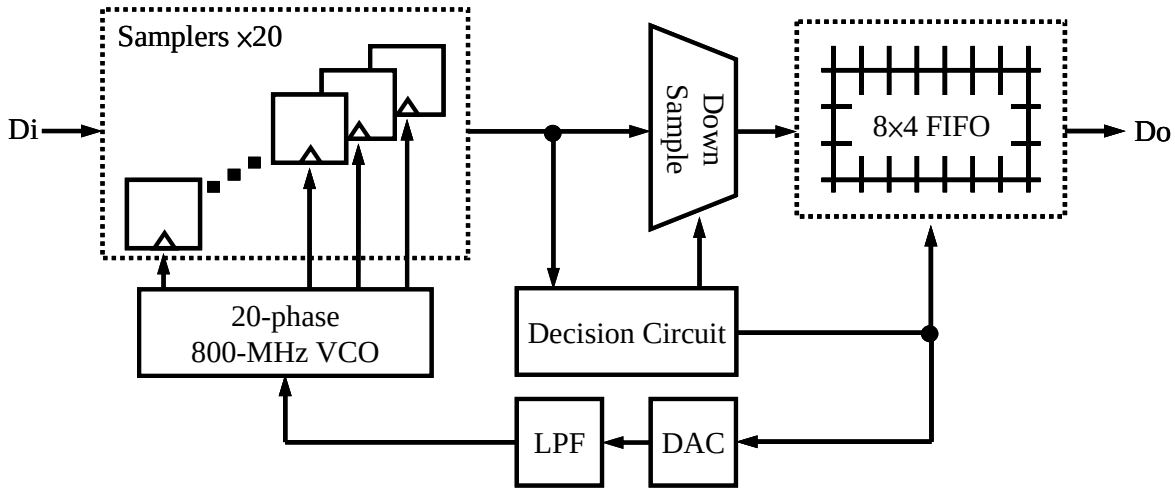



Fig. 2.15: Semi-blind oversampling CDR, Ierssel2006

[2] The original topic is about DLL/PLL instead of CDR.

■ **Hybrid DLL/PLL CDR, T.Lee92** [24]

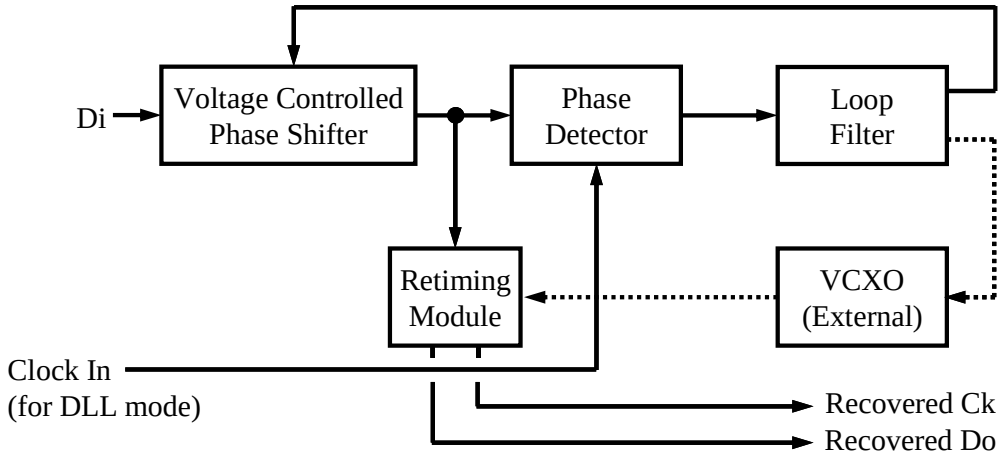



Fig. 2.16: DLL & DLL/PLL CDR, T.Lee92

■ **Hybrid DLL/PLL, Bae&Wei2004** [25]

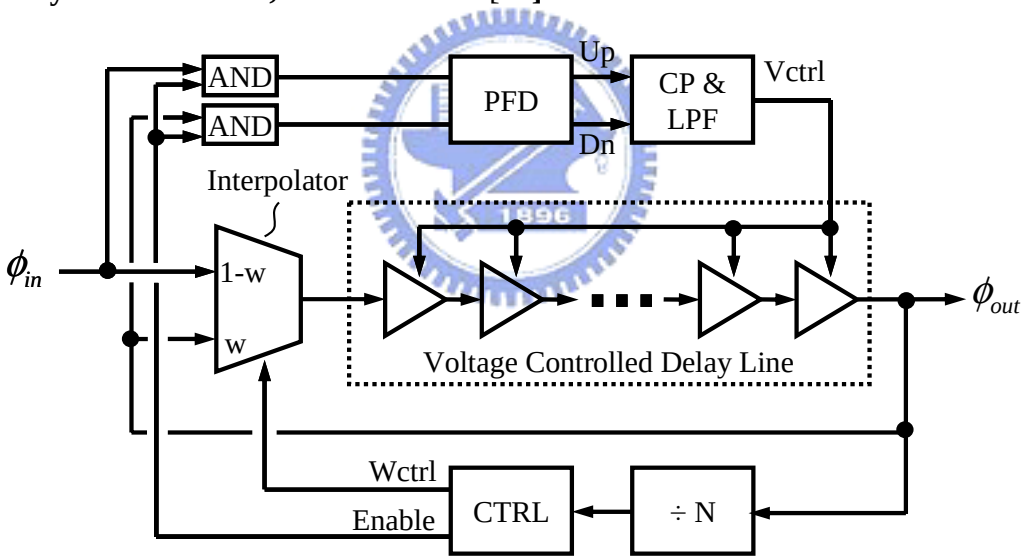



Fig. 2.17: Mixed PLL/DLL, Bae&Wei2004

# 2.6 Summary

Table 1 shows the summary on the CDR architectures, where ○ denotes yes, △ for partially yes, and X for no. As for the blank area, it is a currently un-explored field in this survey. Take the lack of digital implementation of Gated-VCO for example; fast analog circuits are required for the multi-gigabit timing recovery. And

even in today's 10-Gb/s data rate in 0.13-μm CMOS technology, the analog circuits in [20] still employ passive inductors to overcome the bandwidth limitation.

Table 1: CDR architectures summary

| CDR Architectures | | Operation Mode | | Implementation | | Multi-channel Application |
|---|---|---|---|---|---|---|
| | | Cont. | Burst | Analog | Digital | |
| PLL-Based | | ○ | | ○ | △ | |
| Blind Oversampling | | ○ | ○ | △ | ○ | ○ |
| DLL-Based | a) Ck-Interpolation | ○ | ○ | ○ | ○ | ○ |
| | b) Data-Deskew | X | ○ | ○ | ○ | ○ |
| Gated-VCO | | | ○ | ○ | | ○ |
| Alternative & Hybrid | FSM-Based | | ○ | | ○ | |
| | Oversampling/PLL | ○ | ○ | △ | △ | |
| | DLL/PLL | ○ | ○ | ○ | | |

This thesis adopts the data-deskew CDR architecture for 10-Gb/s/ch timing recovery, since a simple architecture is suitable for digital-circuit implementations. The chip is fabricated in 0.13-μm CMOS technology, and operates at a single 1.2-V supply.

# Chapter 3

# Deskew CDR:

# System-Level Analysis

Design of the CDR follows a top-down design concept. The burst-mode multi-channel application introduces the system specifications as well as the architecture candidates. To implement with digital circuits, a DLL-based data-deskew CDR architecture is adopted. Once the architecture is determined, the details on system behavior can be figured out, and the design parameters can be derived.

For “specifications-to-architecture” design, this section starts from the system architecture to provide a clear picture of the CDR. It is then followed by system specifications, design parameters, and the multi-channel synchronization behavior. Both behavioral and mathematical analyses are conducted. After that, the CDR is built and verified with the behavioral models in Simulink.

## 3.1 System Architecture

The CDR targets at on-chip burst-mode timing recovery. It cooperates with an 8-phase PLL at the receiver site. Odd-phase clocks align to the edges of the input data.

They are recognized as *the aligning clocks*. Even-phase clocks are used as *the sampling clocks*.

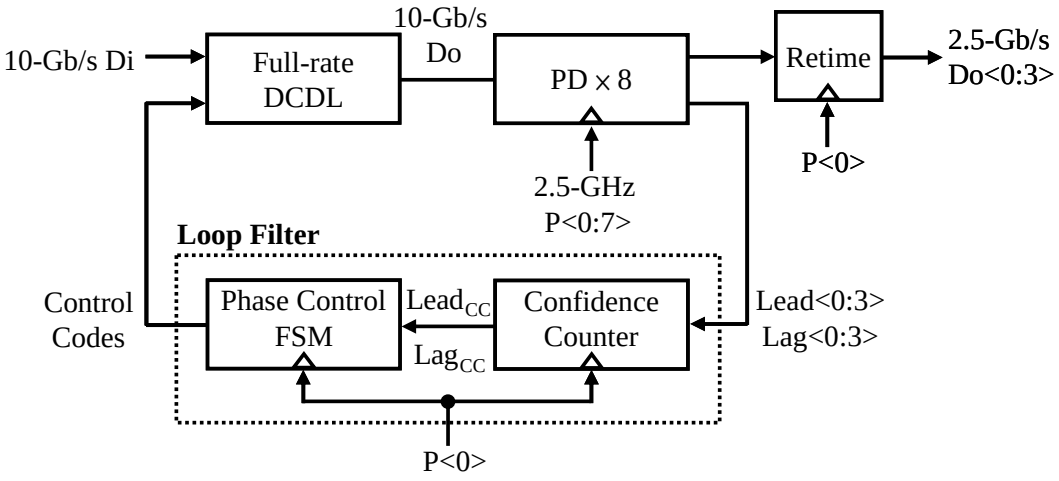



Fig. 3.1: Deskew CDR system architecture

Fig. 3.1 shows Deskew CDR system architecture. It consists of 1) a full-rate DCDL, 2) phase detectors, 3) a confidence counter, 4) a finite state machine, and 5) the retiming stage.

DCDL adjusts the delay of the input data to obtain the optimal sampling phase. It aligns the mid-point of data period to the sampling clock. The full-rate circuit operation occurs only in DCDL. All circuits elsewhere operate in the quarter-rate domain.

Phase detectors are composed of eight flip-flops. They use the 8-phase quarter-rate clocks from PLL at RX. They are configured as 4 sets of Alexander phase detectors (APD), so 4 sets of Lead/Lag information are available from this block. The term Lead/Lag implies that *the data leads/Lags the clock*.

The digital low-pass loop filter is composed of a confidence counter and a phase control FSM. It emulates a single-pole R-C filter in continuous time domain but operates in discrete time domain instead. The closed loop bandwidth is determined by the counter size **N**.[3]

The confidence counter accumulates the 4 sets of Lead/Lag information. If the counting value exceeds the limitation of N/-N, the output $\text{Lead}_{CC}$ / $\text{Lag}_{CC}$ will be produced. FSM updates its state according to $\text{Lead}_{CC}$ / $\text{Lag}_{CC}$ from the confidence

[3] The counter size N refers to the 10-Gb/s domain, even though the confidence counter operates in 2.5-GHz domain. For the case of N = 24, it implies that the minimum *phase update time* is 24-bit time or 2.4 ns.

counter. The current state of FSM determines the control codes of DCDL.

By adding or removing delay, the CDR will finally come to its lock state. In phase detectors, the data is recovered and inherently de-serialized into parallel data. Then, the retimed four 2.5-Gb/s data are available from this system.

## 3.2 System Specifications

Table 2: Specifications of the 10-Gb/s Deskew CDR

| | |
|---|---|
| Burst-data length | 1200 bit |
| Frequency tolerance | ± 500 ppm |
| Peak-to-peak clock jitter between TX and RX | 0.4 UI |
| Tuning range of DCDL | 1.4 UI |
| Phase resolution of DCDL | 6 ps |
| DCDL-induced Jitter | 0.1 UI |

In Table 2, the length of the data packet is 1200-bit time. It includes a preamble of 176 bits for the initial clock recovery. The tuning range of DCDL is 1.4 UI, where the 1-UI delay is for compensating static phase offset and the 0.4-UI delay is for tracking the frequency error.

According to the frequency tolerance specification, the CDR is able to handle the frequency error of ±500 ppm. Or, the CDR can respond to the certain frequency error as long as DCDL does not overflow. In our definition, this is a dynamic specification to the CDR.

Since the data-deskew type CDR has a finite tuning range, the peak-to-peak clock jitter specification should be considered together with the frequency tolerance specification. The clock jitter specification confines the total accumulated phase error to 0.4 UI during each burst period. It guarantees that DCDL will not overflow.

## 3.3 Design Parameters

### 3.3.1 Acquisition Time

Acquisition time is the time to obtain optimal sampling phase. It is the initial tracking time of the CDR. To simplify the analysis, two assumptions have been made.

First, assume the frequency error between TX and RX is small enough and can be neglected. Second, assume that phase detectors give correct Lead/Lag information.

When the system is just initialized, the input data requires some delay to align to the sampling clock. For the best case, the delay is zero. It just aligns to the correct phase at the first place. As for the worst case, it requires the delay of half UI, or 50 ps. According to the fact of probability, the average required delay is ±25 ps or 25 ps in the absolute value. Fig. 3.2(a) shows the timing diagram of the average acquisition.

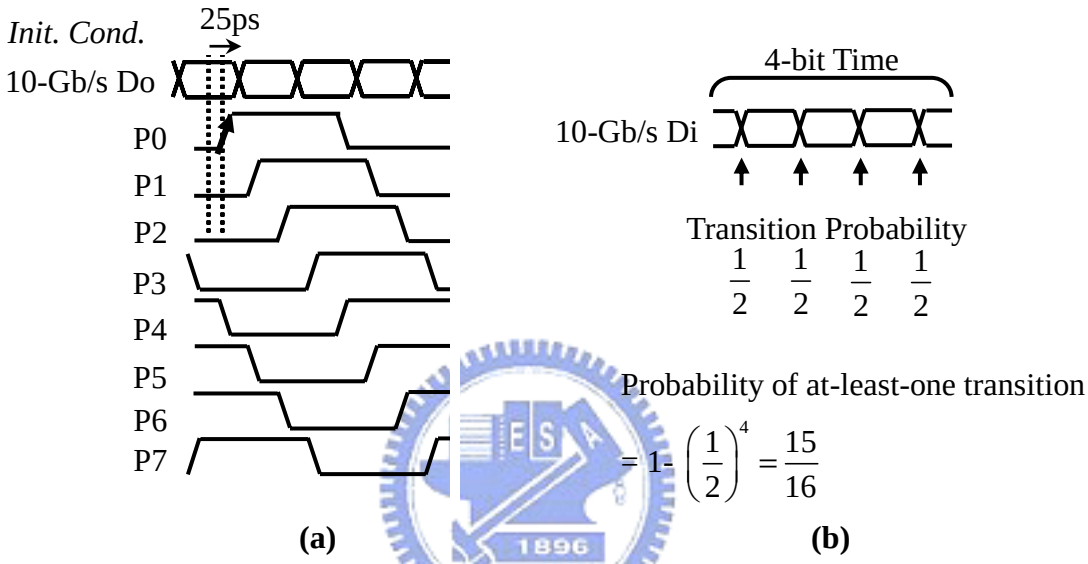



Fig. 3.2: (a) Avg. required delay (25ps) (b)4-bit time transition probability

Assume the edge density ***d*** for one-bit time is 0.5. So, the transition probability for continuous 4-bit time is $(1-d^4) = 15/16$, shown in Fig. 3.2(b). The average acquisition time of this system is

$$\text{Avg. Acquisition Time} = \frac{25\text{ ps}}{\text{Avg. Resolution}} \times \frac{\text{Phase Update Time}}{\text{4-bit-time Transition Probability}}. \quad (3.1)$$

Consider the counter size N = 24, average phase resolution is 6 ps, and phase update time is $N \times 1\text{ UI} = 2.4\text{ ns}$. By substituting them into (3.1), the average acquisition time is 10.67 ns, or 107-bit time. As for the worst acquisition time, it is double the previous value or 21.33 ns, or 213-bit time.

### 3.3.2 Loop-Latency Constraint

The term *loop latency* sums up the circuit operation time through the whole loop.

It is the total process time of the loop. The term *loop-latency constraint* is

$$\text{Loop Latency} < (N \times 1\ \text{UI}). \tag{3.2}$$

where N is the counter size. The loop-latency constraint has to be considered for any closed-loop system. If this constraint is not met, the out-of-date Lead/Lag decisions will be accumulated in the loop. The unstable locking behavior may degrade the performance of the CDR or even cause oscillation.

### 3.3.3 Frequency Tolerance

The frequency tolerance is the maximum frequency error between TX and RX that a CDR can deal with. The frequency tolerance depends on what noise profile is adopted. This analysis starts with an introduction to two types of noise profiles. The first one has the uniform distribution and the second one has Gaussian distribution, which represent the worst and the best cases respectively. [27]

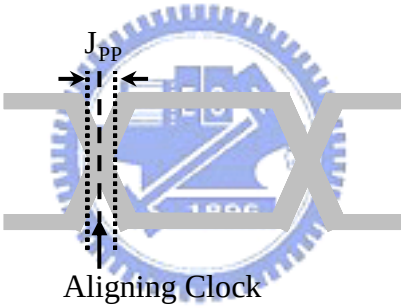
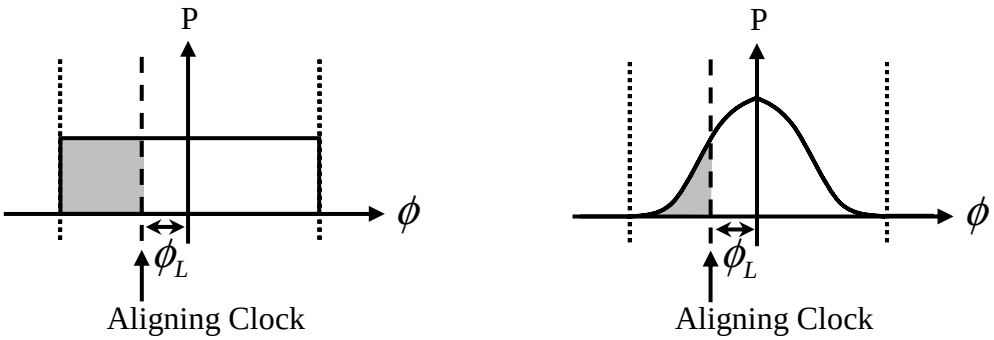



Fig. 3.3: Data eye diagram with the sampling clock




**(a)** **(b)**

Fig. 3.4: Jitter models of a) the uniform distribution b) Gaussian distribution

Consider the CDR operates at lock state. The aligning clock aligns to data edge with a phase error. Fig. 3.3 shows the data eye diagram with the aligning clock. In Fig. 3.4, the phase error is denoted as $\phi_L$. It is the phase offset from the center of the jitter

distribution.

The probabilities of Lead/Lag are functions of $\phi_L$. In this work, Lead and Lag control the Up and Down counting of the confidence counter.[4] Their probabilities are denoted as $P_u$ and $P_d$.

Fig. 3.4(a) shows the noise profile of the uniform distribution. The probability of Up in the gray area is

$$P_u = \frac{J_{pp}/2 - \phi_L}{J_{pp}}. \tag{3.3}$$

To derive the frequency tolerance, firstly discover the net probability of Lag to that of Lead. The net probability of Down to that of Up is

$$P_d - P_u = (1 - P_u) - P_u = \frac{2\phi_L}{J_{pp}}. \tag{3.4}$$

Fig. 3.4(b) shows the noise profile of Gaussian distribution. The probabilities of this noise profile can be expressed as

$$P_u = Q\left(\frac{\phi_L}{J_{rms}}\right), \tag{3.5}$$

$$P_d - P_u = 1 - 2 \cdot Q\left(\frac{\phi_L}{J_{rms}}\right). \tag{3.6}$$

According to (3.4) and (3.6), frequency tolerances of the uniform and Gaussian distributions are derived in (3.7) and (3.8) respectively.

$$\frac{\Delta f}{f} = \frac{d}{N \cdot L}\left(\frac{2\phi_L}{J_{PP}}\right), \tag{3.7}$$

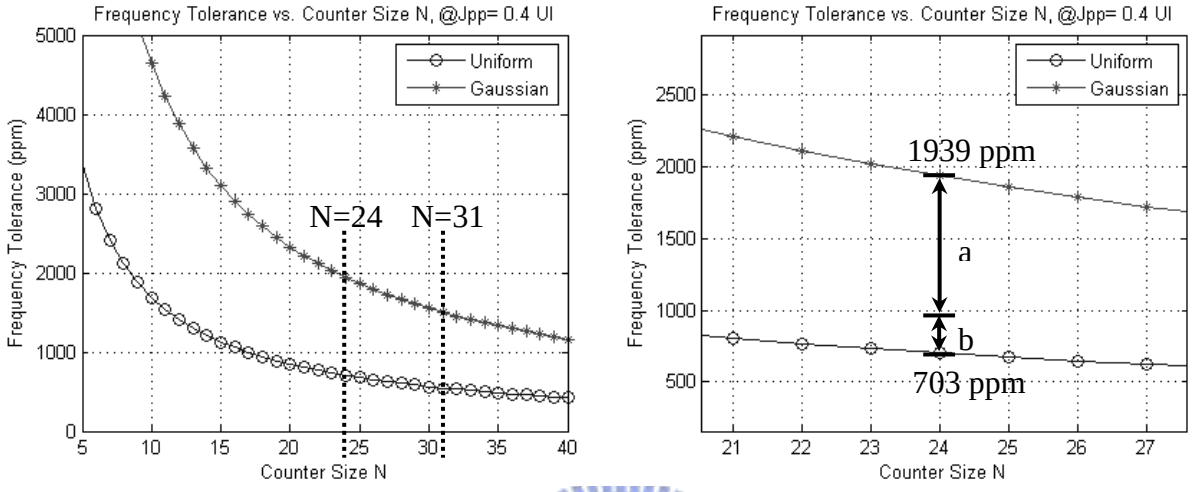
$$\frac{\Delta f}{f} = \frac{d}{N \cdot L}\left(1 - 2Q\left(\frac{\phi_L}{J_{rms}}\right)\right). \tag{3.8}$$

where $\Delta f$ is the frequency offset. $L$ is the number of steps per unit interval. The value of $L$ is

$$L = \frac{\text{Data Bit Time (1UI)}}{\text{Phase resolution}} = \frac{100\text{ ps}}{6\text{ ps}} \sim 16.67. \tag{3.9}$$

[4] Lead/Lag indications result in Up/Down counting behaviors. They also refer to “adding delay to data/removing delay from data” in this work.

To simplify the simulation, certain assumptions have been made. 1) $\phi_L$ = phase resolution = 0.06 UI, 2) the equivalent edge density = $1-(1/2)^4 = 15/16$, and 3) $J_{PP} = 0.4$ UI, $J_{rms} \approx 0.04$ UI. The simulation result of the frequency tolerance is shown in Fig. 3.5.




**(a)** **(b)**

Fig. 3.5: Simulation of frequency tolerance (a) the main plot (b) a zoom-in version

### 3.3.4 Counter Size

The frequency tolerance is a function of the counter size N. The larger N results in the narrower bandwidth. In this case, the high-frequency noise can be filtered out more but at the cost of slower response for the acquisition. On the contrary, the smaller N can handle a larger frequency error. It gives a faster response for the acquisition time.

To determine N, firstly consider the specification of the frequency tolerance. It is ± 500 ppm or 1000 ppm in the tolerant range. Assume a noise profile has both the uniform and Gaussian distributions, and the ratio of the proportions is 50% to 50%. In Fig. 3.5(a), the **maximum N** for handling the 1000-ppm frequency error equals to 31. Since the confidence counter operates in 2.5-GHz domain, the loop filter updates its information every 4-bit time in 10-Gb/s domain. The candidates of N should be the multiples of 4, such as 28, 24, 20, etc.

The **minimum N** is introduced to the system by the loop-latency constraint. Even in the pseudo-NMOS circuit implementation, the process time of the entire loop

requires at least 6 cycles of the 2.5-GHz clock. The candidates of N are 24 or 28 now. To achieve faster acquisition, the smaller N is chosen.

In turn, the decision of N = 24 can derive an equivalent noise profile. From Fig. 3.5(b), the proportion of the uniform distribution to that of Gaussian distribution equals to ( a : b ), where $a \approx 75\%$, $b \approx 25\%$. The derived noise profile is stricter than the initial 50%-50% assumption.

In summary, the CDR is designed with a counter size N of 24. It is capable of the 1000-ppm frequency tolerance with a noise profile of *75%* uniform and *25%* Gaussian distributions.

### 3.3.5 Loop Bandwidth

The closed-loop bandwidth of the CDR can be approximated and measured by adding a simple sinusoidal jitter to the input [28]. The maximum slope of the sinusoidal input jitter is handled by the CDR. The slew rate of the CDR should be larger than the maximum slope of the sinusoidal input jitter below the loop bandwidth $f_{BW}$.

The relationship between the frequency tolerance $\Delta f / f$ and the slope of the input jitter is

$$\frac{\Delta f}{f} = A_j \cdot (2\pi f_{BW}), \qquad (3.10)$$

$$f_{BW} = \frac{\Delta f / f}{2\pi A_j}.$$

$A_j$ is the jitter amplitude in second in (3.10), and $f_{BW}$ is inversely proportional to $A_j$. So the loop bandwidth decreases as the input jitter amplitude grows. Considering both the noise profiles of the uniform and Gaussian distributions and N = 24, we can derive the relationship of $f_{BW}$ and $A_j$ from (3.7) - (3.10), shown in Table 3.

Table 3: The loop bandwidth $f_{BW}$ vs. the input jitter amplitude $A_j$

| Input jitter amlitude $A_j$ | 0.2 UI | 0.3 UI | 0.4 UI |
|---|---|---|---|
| $f_{BW}$ of uniform distribution | 22.4 MHz | 10.0 MHz | 5.6 MHz |
| $f_{BW}$ of Gaussian distribution | 37.2 MHz | 23.5 MHz | 15.4 MHz |

### 3.3.6 Delay Line Related

**Tuning Range** is the maximum tuning delay that DCDL can afford. Or, it is the difference between the maximum and the minimum data delay. In our system, it is 1.4 UI or 140 ps. The tuning range of 1.4 UI can be divided into two parts. In the worst case, the 1-UI delay is used to obtain the optimal sampling phase. And, the 0.4-UI delay is used to track the frequency error.

**Phase Resolution** is the minimum tuning delay. It is developed by advancing one LSB of the control code. The *average phase resolution* is the tuning range divided by the number of codes. The specification for the average phase resolution is 6 ps.

**Input Sensitivity** is the required minimum input amplitude that the input stage can handle and amplify to a target level. The specification is ± 50 mV.

### 3.3.7 Peak-to-Peak Clock Jitter between TX and RX

The peak-to-peak clock jitter specification is a constraint for both PLLs at TX and RX. During timing recovery, the total peak-to-peak clock jitter has to be confined. Even with the frequency error between TX and RX, the total accumulated phase error in each burst period must follow this specification. The allowed peak-to-peak jitter between TX and RX for the CDR is 0.4 UI.

An equivalent frequency error of 333.33 ppm can be derived from this 0.4-UI peak-to-peak jitter. To simplify the analysis, we assume the frequency error is a frequency step. This frequency step is applied to the CDR at lock state. The accumulated phase error $\Delta t$ in second can be represented as

$$\begin{aligned} \Delta t &= k \cdot \left( \frac{1}{f_0} - \frac{1}{f_0(1+f_{err} \cdot 10^{-6})} \right), \\ &= k \cdot \frac{1}{f_0} \left( 1 - \frac{1}{1+f_{err} \cdot 10^{-6}} \right), \\ &\approx k \cdot \frac{1}{f_0} \cdot \left( f_{err} \cdot 10^{-6} \right). \end{aligned} \tag{3.11}$$

where **k** denotes the bit length of data packet. The full-rate $f_0$ equals to 10 GHz, and $f_{err}$ denotes the frequency error in ppm. Rewrite (3.11),

$$\Delta t = k \cdot f_{err} \cdot 10^{-4} \text{ ps.} \tag{3.12}$$

Substitute $\Delta t = 40$ ps, $k = 1200$ into (3.12). The equivalent frequency error of 333.33 ppm is derived.

# 3.4 Multi-Channel Synchronization

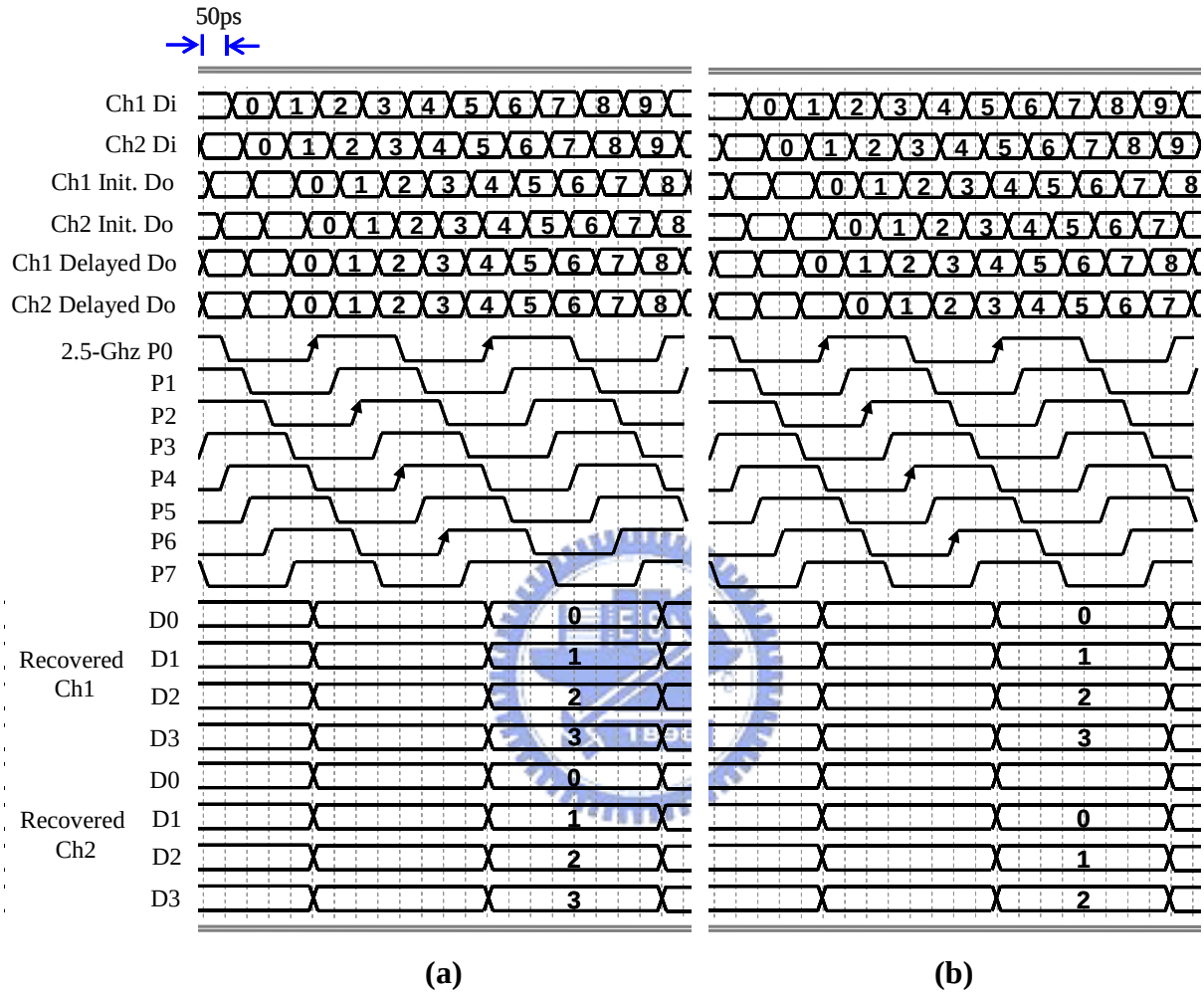



Fig. 3.6: Multi-channel data recovery (a) aligning case (b) shifting 1-bit case

This section depicts the synchronization behavior of multi-channel timing recovery. The recovered data of multiple channels are synchronized to a particular clock phase of PLL. Ideally, the edges of the multi-channel data align to the same odd-phase clock. That is, the recovered first bit of the first channel aligns to the recovered first bit of the Nth channel. But for the practical usage, the edges of the multi-channel data may align to different odd-phase clocks of PLL. It depends on the skews among channels.

Fig. 3.6 illustrates a 2-channel timing recovery. The *Init. Do* is the initial output of DCDL. Before the data acquisition process, the delay of half tuning range is added

to the data in the initial setting. After that, the data delay can be tuned by: 1) adding more delay to the data, or 2) removing the added delay from the data. *Delayed Do* denotes the output data of DCDL when the system comes to its lock state.

In this work, even-phase clocks are the sampling clocks. The 4 recovered data D<0:3> are developed from P<0>, P<2>, P<4>, P<6> respectively. They are sampled by P<0> again. Assume the clock-to-Q delay is zero. The recovered data D<0:3> are derived after two clock period of P<0>.

Fig. 3.6(a) shows the delayed Do of Ch1 and that of Ch2 align to the same clock. So, the recovered data of Ch1 and that of Ch2 are aligned. In Fig. 3.6(b), the delayed Do of Ch1 and that of Ch2 are skewed by 1-bit period. So, the recovered data are shifted by 1-bit position. The parallel data re-arrangement is to be handled at a higher level.

# 3.5 Behavioral Models

This section depicts the behavioral models of Deskew CDR in Simulink. It makes a brief emphasis on what has been considered, how to build the models, and the ways to set up the environment for the simulations.

## 3.5.1 System Architecture

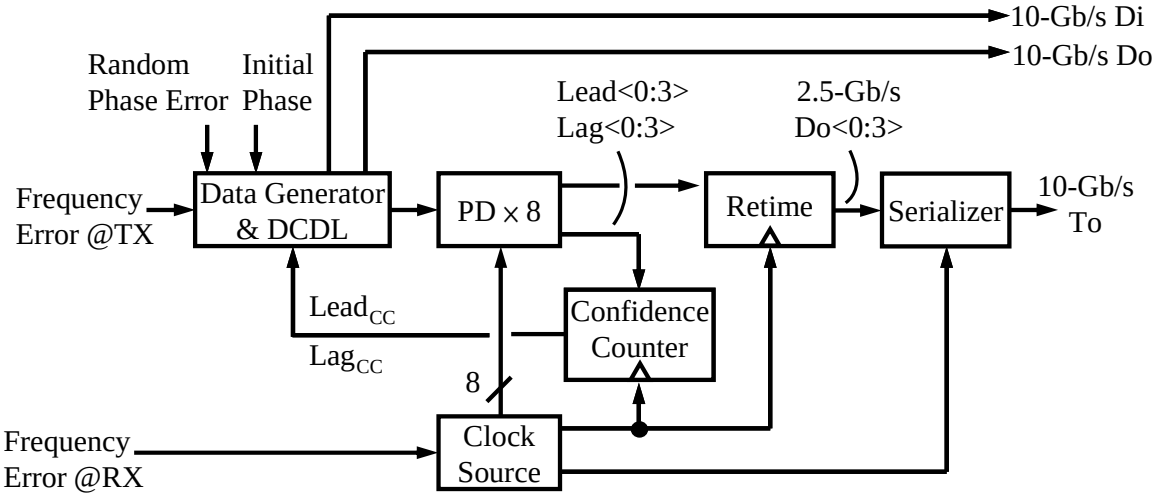



Fig. 3.7: Deskew CDR system architecture in Simulink

Fig. 3.7 shows the CDR architecture in behavioral models. The *data generator*

generates the full-rate PRBS data to DCDL. The *initial phase* of the input data of DCDL is parameterized. And, the *frequency error at TX* is added here.

The *frequency error at RX* is added in the clock source. Besides, an additional block *Serializer* is introduced to the system. It generates the 10-Gb/s output data *To* for the comparison to Di and Do, where Di is the input data of DCDL, and Do is the output data of DCDL.

### 3.5.2 Data Generator & Delay Line

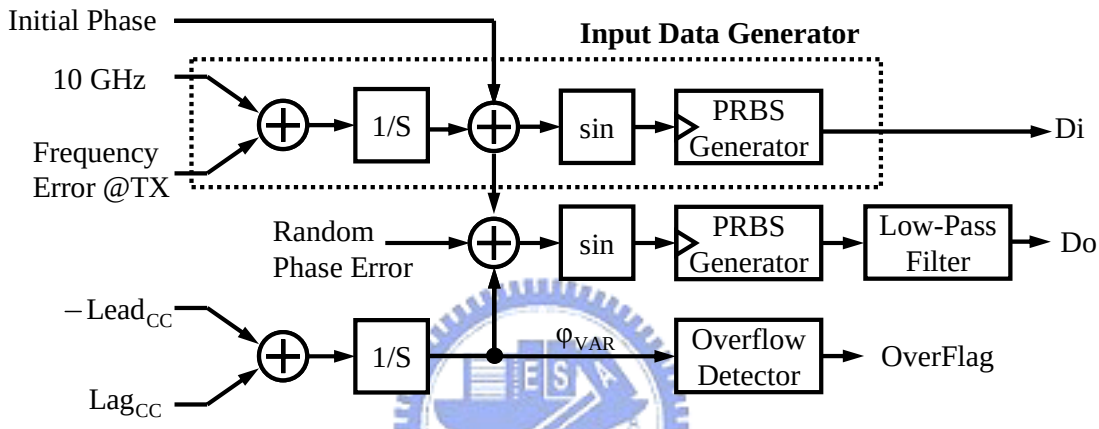



Fig. 3.8: Data generator & DCDL in Simulink

In Fig. 3.8, there are two signal paths, Di's path and Do's path. *Di* is the input of DCDL. *Do* is the delayed and noisy output of DCDL.

For Di's path, the 10-GHz frequency adds the *Frequency Error at TX*. Through the integrator *1/S*, the frequency is converted to a phase. A sinusoidal function, denoted as the *sin* block, transfers this phase into a clock in time domain. The PRBS generator uses this clock and generates Di with the frequency error of TX. The initial phase of Di is parameterized by *Initial Phase*. Note that Di is not actually fed to DCDL in this configuration, but it does model the input data of DCDL.

For Do's path, it begins from the input $\text{Lead}_{CC}$ and $\text{Lag}_{CC}$ to the output Do. The tuning behavior of DCDL is modeled by adjusting the clock phase of the PRBS generator. It is based on the fact that *Adding delay to data* is equivalent to *removing delay from the clock*.

In the original design, DCDL adds one phase resolution to the data when $\text{Lead}_{CC}$ validates. As for this configuration, the clock should subtract one phase

resolution. So, in Fig. 3.8, the input $\text{Lead}_{CC}$ is together with a negative notation. $\text{Lag}_{CC}$ has the similar function as $\text{Lead}_{CC}$. The phase is then added with a random phase error. Through sin block, the phase is converted to the clock. The following PRBS generator synchronizes to this clock. It generates a random pattern in the polynomial of $(1+x^3+x^{31})$. Finally, the data pattern goes to the *Low-pass Filter* block, so the ISI effect is also modeled. The implementation of this block includes a low-pass transfer function and a hysteresis comparator.

Besides these two signal paths, there is a path for the overflow indication, *OverFlag*. The tuning range specification of DCDL is 1.4 UI. OverFlag validates when the accumulated tuning phase $\varphi_{VAR}$ is above 0.7 UI or below -0.7 UI.

### 3.5.3 Noise Sources

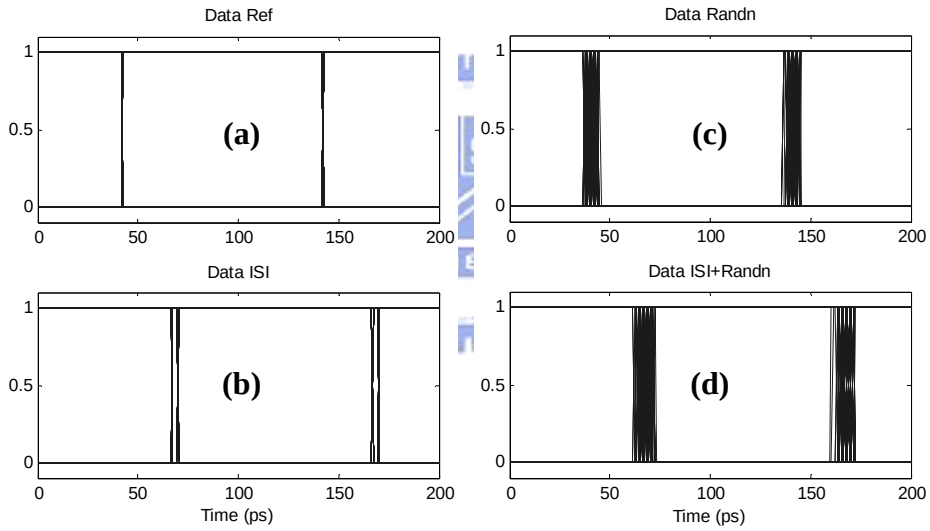



Fig. 3.9: Adding noise sources in Simulink (a) a reference eye (b) ISI effect (c) random noise (d) the combination of ISI and random noise

To simply observe how the noise profiles affect the eye diagram, the path of $(\text{Lag}_{CC}\text{-Lead}_{CC})$ in Fig. 3.8 is temporarily removed. Do is the output of DCDL in Fig. 3.8. Fig. 3.9 shows the eye diagrams of Do.

In Fig. 3.9, case (a) shows the ideal eye diagram of Do. In case (b), it shows the eye diagram with ISI effect. Comparing to case (a), the eye is delayed due to the low-pass transfer function. The deterministic jitter is added. In case (c), it shows Do with random phase noise. A $\pm 4\sigma$ Gaussian model is applied. In case (d), it shows

the combination of both ISI and random phase noise.

For those eyes in Fig. 3.9 except the reference one, their jitter histogram plots are given in Fig. 3.10.

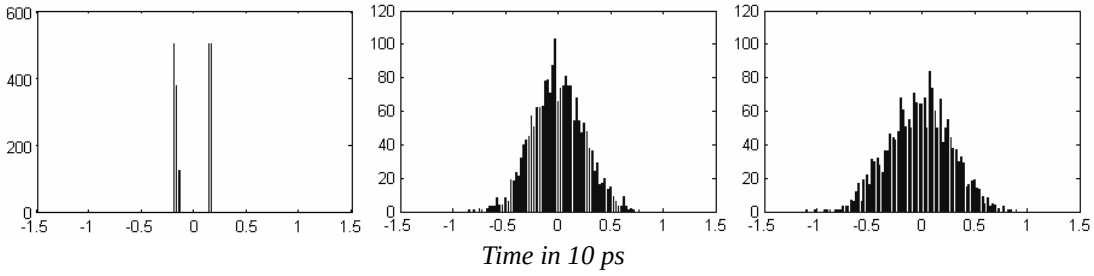



Fig. 3.10: Jitter histogram plots of (a) ISI (b) Random noise (c) ISI and Random noise

### 3.5.4 Phase Detectors

Fig. 3.11 shows eight DFFs serving as the phase detectors. P<0:7> are the sampling clocks. The edge spacing for each adjacent clock is 50 ps. Every 3 DFFs and 3 XORs compose a set of APD. There are 4 sets of APD in this block. Therefore, ther are 4 sets of Lead/Lag outputs.

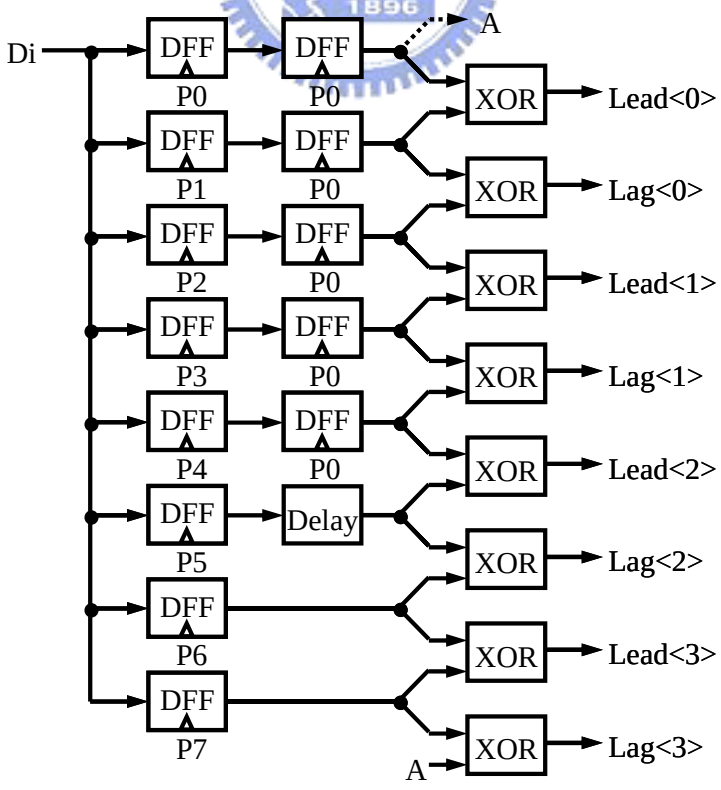



Fig. 3.11: Phase detectors' block in Simulink

In phase detectors' block, *Clock to Q Delay* $T_{CQ}$ is critical. All the even-phase clocks align to the midpoint of the data period. They sample the intput Di at the optimal phase. So, $T_{CQ}$ of even-phase detectors is 50 ps. The odd-phase clocks align to the transition edges of the input Di. The $T_{CQ}$ is degraded to 125 ps. Behaviors of the circuit-level $T_{CQ}$ determine the retiming scheme for APD, and thus need special cares.

### 3.5.5 Confidence Counter

The *confidence counter* and the *phase control FSM* compose the digital low-pass filter. The filter contributes a pole to the CDR. But for the configuration shown in Fig. 3.7 and Fig. 3.8, the FSM is modeled by the integrator 1/S in DCDL. The tuning phase of DCDL is derived instantaneously from $Lead_{CC}$ / $Lag_{CC}$ instead of the control codes of FSM. So, there is not a specific block named FSM in Fig. 3.7.

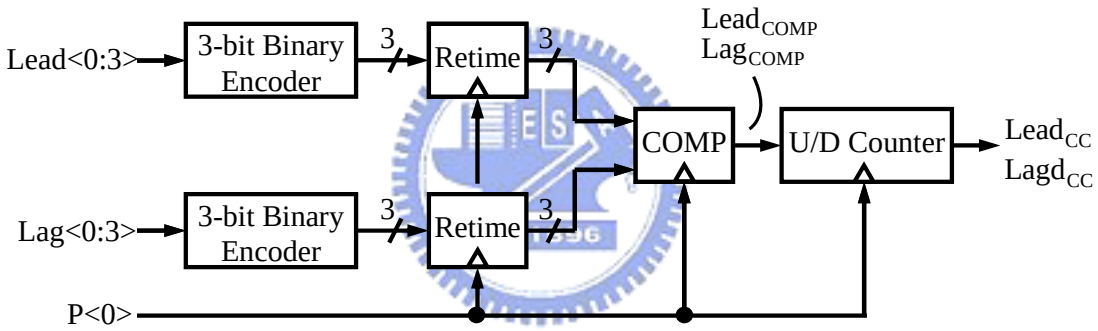



Fig. 3.12: Confidence Counter in Simulink

In Fig. 3.12, the count of Lead<0:3> is encoded into 3-bit unsigned integer, and so does that of Lag<0:3>. After retiming, the integers are compared in *COMP* block. The comparison determines which integer is the larger one. $Lead_{COMP}$ will be true if the encoded integer from Lead<0:3> is the larger one.

The comparison result $Lead_{COMP}$/$Lag_{COMP}$ goes to the *U/D Counter* for further accumulation. If the counting value reaches the maximum/minimum limitations, $Lead_{CC}$ / $Lag_{CC}$ will be validated. Then, the counter will be reset to its initial state immediately.

The comparator COMP introduces the **majority-vote** scheme. For the comparing cases of (4:0), (3:1), (3:0), (2:1), (2:0) or (1:0), they all regard the larger integer as **4**. The counting limitations of the final counter are +6/-6, and the equivalent counting

values are +24/-24 respectively. The equivalent counting values refer to the whole confidence counter's size N = 24 mentioned previously. The equivalent edge density is also enhanced by this scheme.

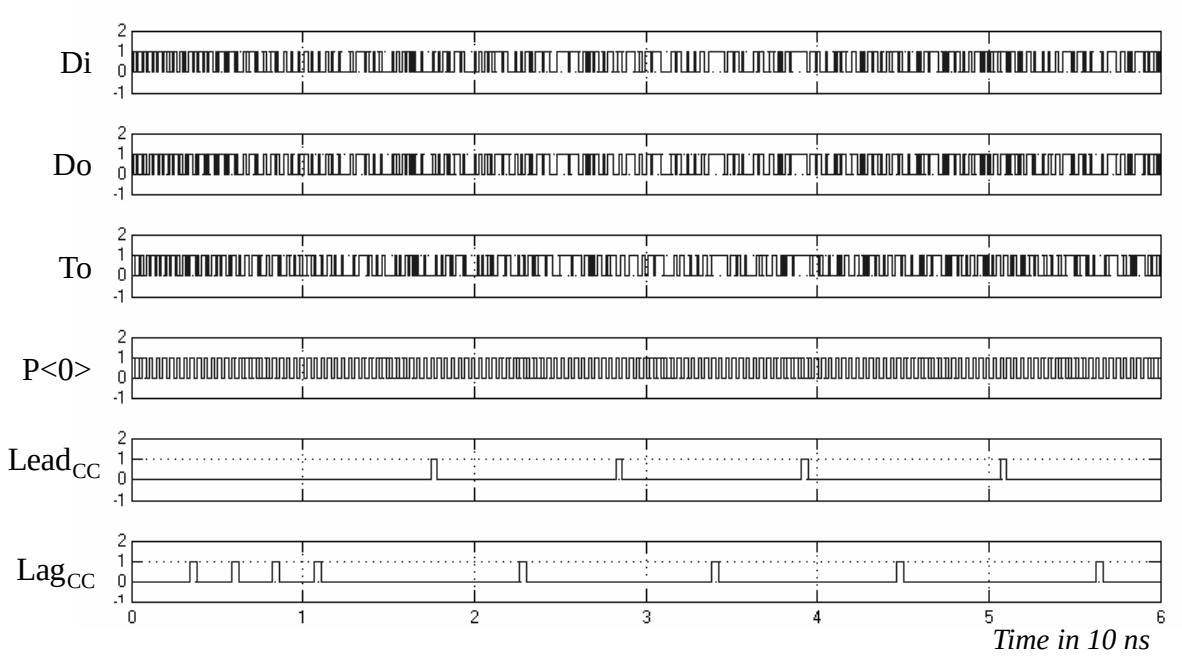
# 3.6 Behavioral Simulations

This section gives demonstrations of the full-system simulation. *Di* is the 10-Gb/s input data of DCDL. *Do* is the delayed output data of DCDL. To is serialized from the parallel recovered data for comparison. *P<0>* is the 2.5-GHz system clock.

## 3.6.1 Random Phase Noise

Fig. 3.13 gives an average case of the required *25-ps* tuning delay, i.e., it requires 4 phase steps to obtain the optimal sampling phase. At the lock state, it shows how the Gaussian noise affects the locking behavior. In this simulation, ISI effect is intrinsic inside DCDL. The amplitude is 0.1 UI.

Fig. 3.13 (a) shows the case without Gaussian noise as the reference case. Fig. 3.13 (b) introduces 0.2-UI Gaussian noise. At lock state, the frequency of the Lead/Lag in (b) is lower than that in (a). It implies that the loop bandwidth is reduced. Fig. 3.13 (c) introduces 0.4-UI Gaussian noise. The system locks to another phase state, which is different from (a) and (b), due to the large random noise.




**(a)**

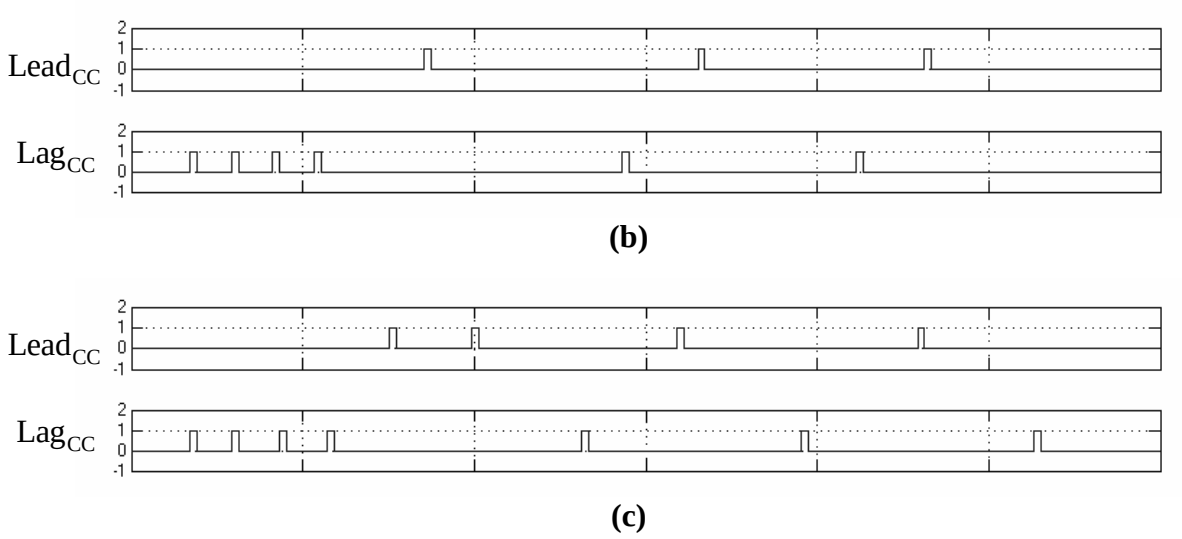



Fig. 3.13: Simulations of random noise (a) free of random noise (b) 0.2-UI random noise (c) 0.4-UI random noise

### 3.6.2 Loop Latency

To observe how the loop latency affects the systematic locking behavior. First, remove the noise sources from the system. So that, decisions of phase detectors are made from the precise data. Second, apply a periodic input to the CDR. Because, the maximum Lead/Lag decisions occur at the maximum edge density of the input data. These two conditions introduce the worst case that breaks the loop-latency constraint.

Fig. 3.14 shows the simulations of the loop latency in four cases

(a) loop latency $<$ (1N $\times$ 1 UI)

(b) (1N $\times$ 1 UI) $\leq$ loop latency $<$ (2N $\times$ 1 UI)

(c) (2N $\times$ 1 UI) $\leq$ loop latency $<$ (3N $\times$ 1 UI)

(d) (3N $\times$ 1 UI) $\leq$ loop latency $<$ (4N $\times$ 1 UI)

For the ease of explanation, Lead and Lag is denoted as A and B. Case (a) is exactly the definition of the loop-latency constraint. The system locks at a continuous A-B-A-B state since the constraint is met. In case (b), the system locks at the A-A-B-B state when the latency is just over the constraint. The more latency will introduce the more instability to the system. As shown in case (c), it locks at the A-A-A-B-B-B state, and case (d) locks at the A-A-A-A-B-B-B-B state.

It is obvious that an extra latency ($N \times 1$ UI) introduces an extra repeating pattern (A-B) to the lock state. Because, the extra latency allows those out-of-date Lead/Lag decisions to be trapped and released in the loop later.

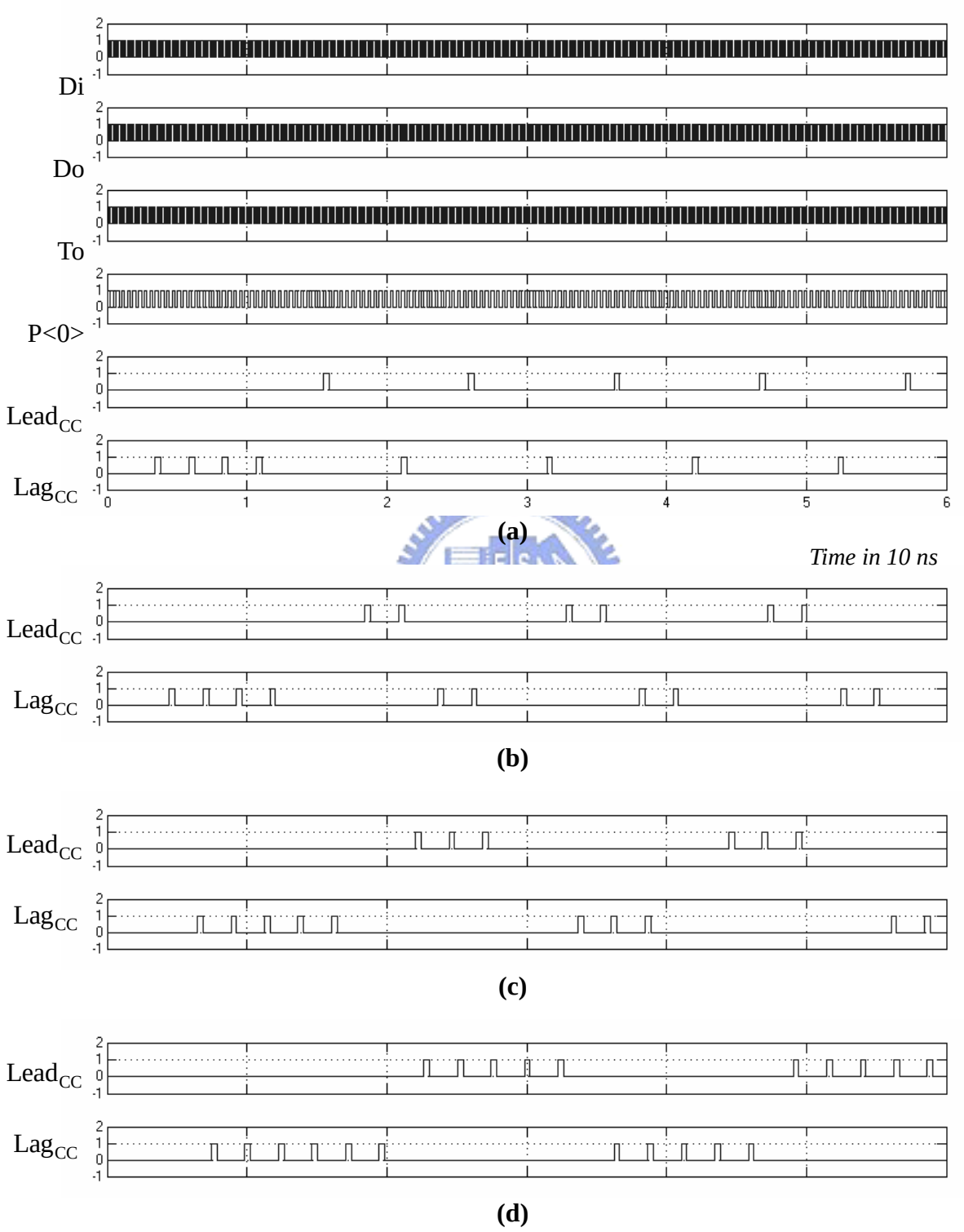


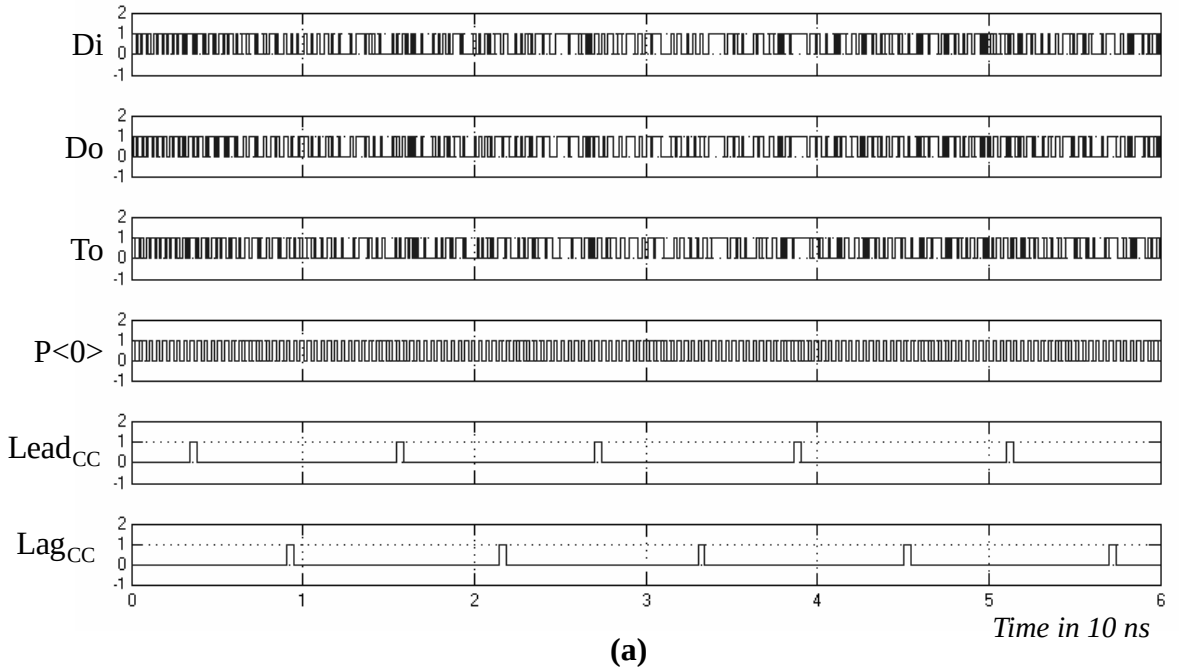

Fig. 3.14: Simulations of loop latency

### 3.6.3 Frequency Error

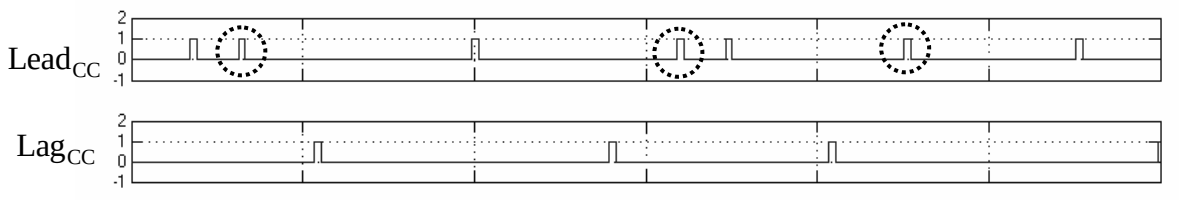
This section depicts how the CDR responds to the frequency error. To simply observe the tracking behavior of the CDR, the delay of the input data is adjusted by the parameter *Initial Phase*. So, the CDR just locks at the origin of the time axis.

Fig. 3.15 shows the simulations of the frequency error in three cases. Case (a) is the reference case. The number of Lead decisions matches that of Lag. It is the ideal case that Lead/Lag always occurs in pair. As for case (b) and (c), there are extra Lead decisions circled by the dashed lines. Those extra Lead decisions tracks the frequency error of TX.

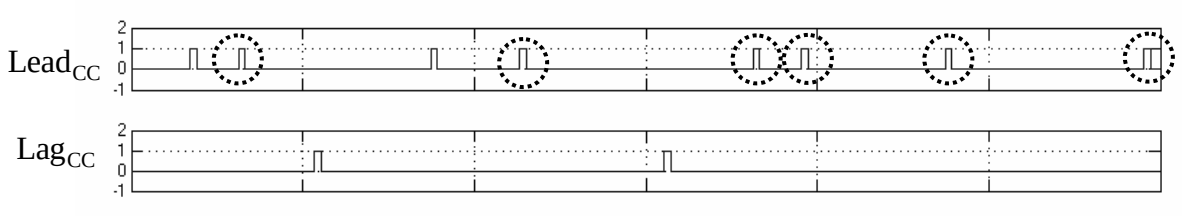
The simulation time is 60 ns, or 600-bit time. From (3.12), it requires a *18-ps* tuning delay to compensate the 300-ppm frequency error lasting for 600-bit time. As for the 500-ppm frequency error, it requires a *30-ps* tuning delay. Since the phase resolution is 6 ps, the extra count of Lead decisions is expected to be *3* for case (b) and *5* for case (c). The simulation shows almost the same as the derived.




**(a)**




**(b)**




**(c)**

Fig. 3.15: Simulations of frequency error (a) free of frequency error, (b) 300-ppm frequency error at TX, (c) 500-ppm frequency error at TX.

So far there is no case that demonstrates the overflow of the delay line. For this phenomenon, the simulation in Fig. 3.16 is created.

In this simulation, the frequency error of TX is +500 ppm, and that of RX is -500 ppm. So, the total frequency error is 1000 ppm. The simulation time is 130 ns or 1300-bit time. According to (3.12), it requires a tuning delay of 130 ps to compensate the 1000-ppm frequency error. However, the maximum upward tunable delay is half of the tuning range or 70 ps. Since $70\text{ ps} / 6\text{ ps} \approx 11.67$, *OverFlag* validates at the *12th phase step* shown in Fig. 3.16.

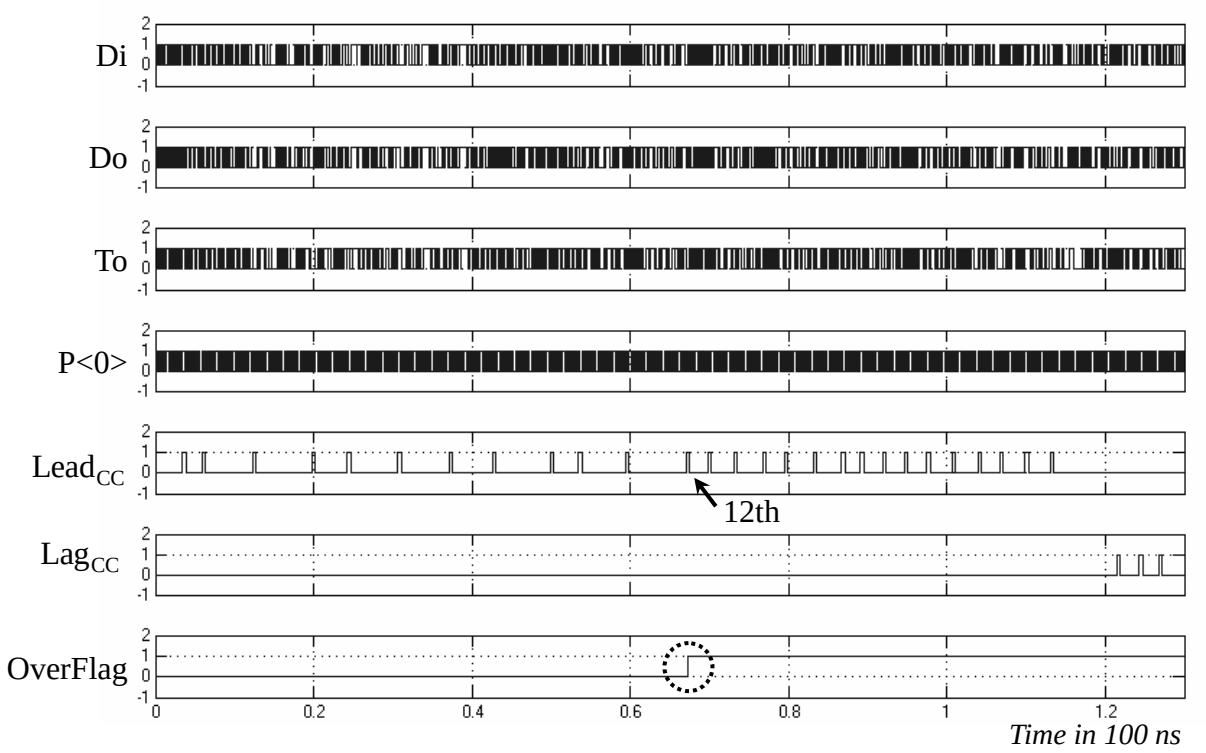



Fig. 3.16: Simulation of DCDL's overflow

Besides, the time when the overflow occurs is predictable. Rewrite (3.12), it becomes

$$k = \frac{\Delta t / ps}{f_{err} \cdot 10^{-4}}. \qquad (3.13)$$

When $\Delta t = 70\ ps$, $k = 700$ is derived. It indicates that DCDL shall overflow at the 700th bit time or 70 ns. In Fig. 3.16, it happens at 67 ns.

After the overflow, the delay line is no longer upward tunable. Those Lead decisions validate without actual tuning actions at the duration of time = 70 ns ~ 120 ns. After 120 ns, the Lag decisions take place. Because, the accumulated phase error finally flips the Lead/Lag relationship between the clock and the data.

# Chapter 4

# Deskew CDR:

# Circuit Implementation

## 4.1 System Architecture

Fig. 4.1 shows the circuit-level architecture of the CDR. According to the operation rate, building blocks of the CDR operate in two frequency domains. DCDL operates in the 10-Gb/s domain. The others operate in the 2.5-GHz domain.

DCDL receives the 10-Gb/s small-signal data. It amplifies and adjusts the delay of the data. Phase detectors have been configured as 4 sets of APDs. They sample the data from DCDL and judge the Lead/Lag relationship between the data and the clock.

The confidence counter consists of 1) an encoder *ENC*, 2) a comparator *COMP*, and 3) an accumulator *ACCU*. In ENC, the counts of Lead/Lag from APDs are encoded into two 3-bit integers. These two integers are compared in COMP. The comparison result goes to ACCU for the further accumulation. If the accumulating value exceeds the counter size N/-N, the output $\mathrm{Lead_{CC}}$ / $\mathrm{Lag_{CC}}$ validates.

The finite state machine *FSM* receives $\mathrm{Lead_{CC}}$ / $\mathrm{Lag_{CC}}$. It updates the state accordingly. Finally, DCDL receives the control codes of FSM. These codes

determine the delay of the data.

A clock buffer block *BUF* is introduced here. It buffers the clocks P<0:7> from the PLL at RX. It guarantees that all P<0:7> see equal capacitances. BUF also provides the 2.5-GHz *Ck* and *Ckb* to the CDR. They are asynchronously buffered from two clocks of the P<0:7>.

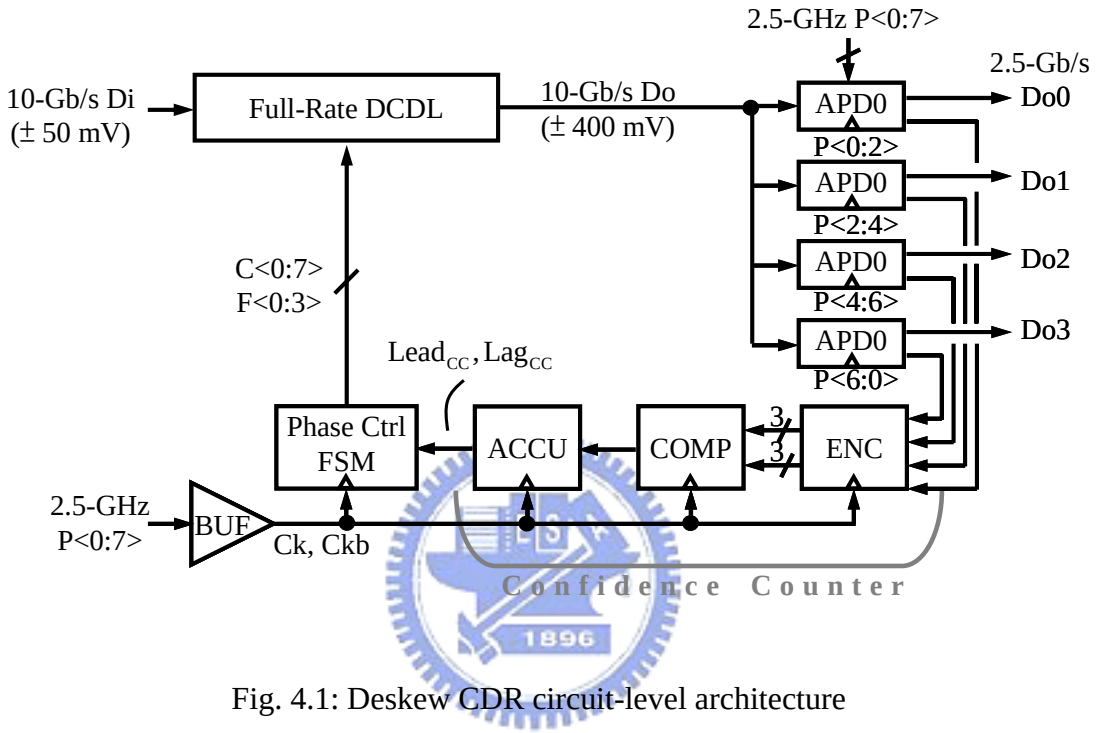



Fig. 4.1: Deskew CDR circuit-level architecture

# 4.2 Critical Design (1): DCDL

Features of *the large tuning delay, good noise margin, and self-adjusted crossing point, etc* are what the large-swing DCDL benefits from the use of the static CMOS inverters. However, the bandwidth of the static CMOS inverter in 0.13-um technology is insufficient to deal with the combination of the input capacitance of the next stage, wiring and other parasitic loads. It is necessary to apply the inductive-peaking scheme to the circuits, so as to achieve 1) the required 18-dB gain at the pre-amplifier stage, and 2) the practical 7-GHz edge frequency[5].

The circuit design of DCDL is critical for two reasons. First is the **insufficient bandwidth** as mentioned previously. Second is the **uncertainty of the parasitic R**

[5] For the 10-Gb/s data rate, one bit time is 100 ps. Consider a practical eye of 50-ps transition time and 50-ps steady time. The 50-ps transition time approximates an edge frequency of 7 GHz.

**and C**. The value of R and C determines the phase resolution of DCDL. It also determines the amount of the inductive peaking. So, several iterative processes between the post simulation and the layout modification are required for the design of DCDL.

### 4.2.1 DCDL Architecture

Fig. 4.2 shows the DCDL architecture. It consists of 1) *a pre-amplifier stage* and 2) *the main delay line*. The pre-amplifier stage is composed of three cascaded cells denoted as *A*. It serves as a limiting amplifier, and provides a total gain of 18 dB to the input data. The swing of the 10-Gb/s data is amplified to its upper and lower bounds, so that the monotonic and smooth tuning per phase step is guaranteed.

The main delay line consists of 8 delay cells denoted as $\Delta t$, where the last delay cell is the dummy one. The notation $\Delta t$ also refers to the propagation delay of a delay cell. There are 7 timing intervals between the 8 delay cells. So, the **tuning range** of DCDL approximates $(7\ \Delta t)$.[6]

Fig. 4.3(a) shows the schematic of the delay cell. The pre-amplifier cell also adopts this schematic. The only difference between these two cells is the strength of the positive-feedback latch. A smaller latch size is adopted by the pre-amplifier cell.

Two tuning functions are available to the main delay line. The **coarse tuning function** is implemented by the propagation delay of a delay cell. In Fig. 4.2, all the data d<0:7> are tapped from the delay line. These data are classified into even/odd types. They are fed to two 4-to-1 multiplexers respectively. Only the adjacent even/odd data pair is selected in the multiplexers. The outputs of the multiplexers are denoted as *Ev/Od*. Fig. 4.3(b) shows the schematic of the multiplexer of the odd-data path. It comprises 4 tri-state buffers. Only one of the buffers validates at a time due to the *one-hot coding* scheme. The multiplexed data pair Ev/Od goes to the interpolator stage.

The **fine tuning function** is implemented by the interpolation of Ev/Od. Fig. 4.3(c) shows the schematic of the interpolator. The fine control *F<0:3>* are in the *thermometer* format. The interpolator determines the proportions of Ev/Od for each interpolation by F<0:3>. The phase difference of Ev/Od is equally divided into 4 portions. So, the **phase resolution** is derived as $(\Delta t/4)$.

[6] The tuning range is $(7\Delta t - \Delta t/4)$ or $(27\Delta t/4)$ because there are totally 28 sets of control codes.

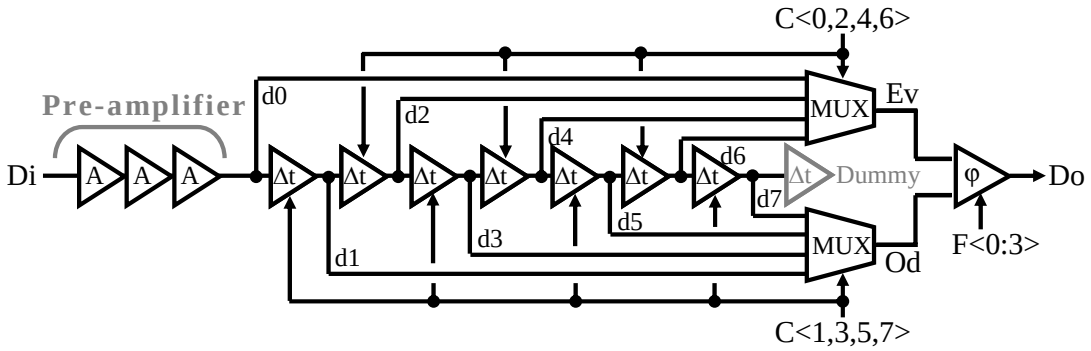



Fig. 4.2: DCDL architecture

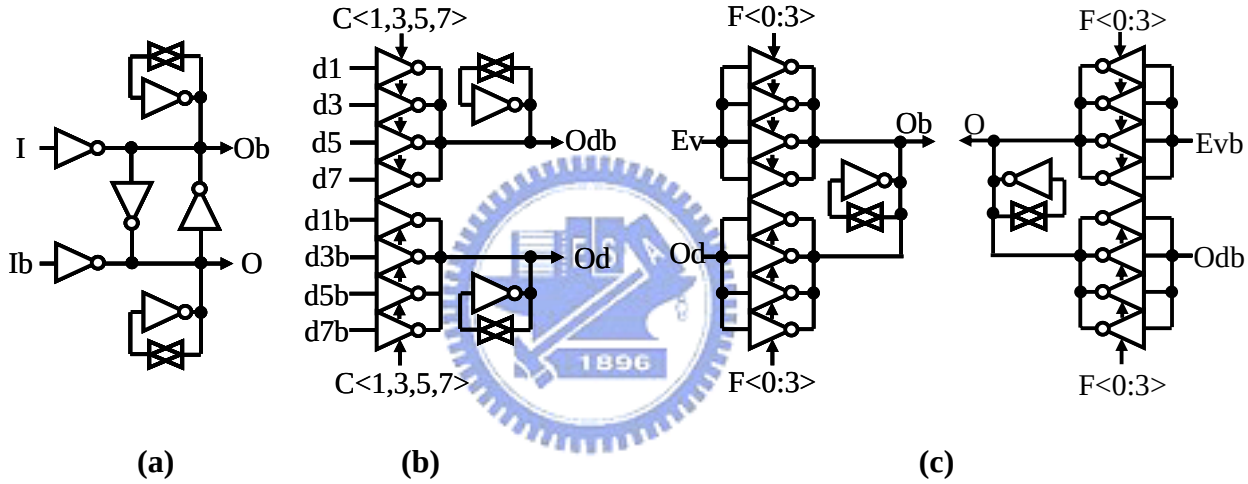



Fig. 4.3: Schematics of (a) a delay cell (b) MUX4:1 (c) the interpolator

### 4.2.2 Inspection by the R-C Model

Consider an ideal step signal is applied to a simple R-C model. The propagation delay of the signal is

$$\ln(\frac{100\%}{50\%})\cdot\tau \approx 0.69\tau \tag{4.1}$$

where $\tau$ is the R-C time constant. The 10%-90% rise time is

$$\ln(\frac{90\%}{10\%})\cdot\tau \approx 2.2\tau \tag{4.2}$$

From (4.1) and (4.2), it follows that

$$\text{Propagation Delay} \approx \frac{\text{Rise Time}}{3} \tag{4.3}$$

To determine *the number of the cascaded delay stages*, we consider the transition time *0.5 UI* of the practical 10-Gb/s eye. From (4.3), the propagation time of one stage approximates $(0.5/3\ \text{UI})$. To achieve the tuning range of 1.4 UI, the required number of stages is derived as $1.4\ \text{UI} / (0.5/3\ \text{UI}) \approx 8.4$. The circuit implementation has 8 stages.

As a rule of thumb, the R-C model provides a rough but usually well-enough estimation. Errors come from the overshooting and/or ringing behaviors of the step response, which are not modeled by an R-C model for sure.

### 4.2.3 Bandwidth Limitation by Technology

Before carrying on the topics of DCDL, we introduce the intrinsic bandwidth limitation by the technology.

Table 4: MOS intrinsic $f_T$ in TSMC 0.13-μm 1P8M technology

| Overdrive Voltage | 200mV | 300mV | 400mV |
|---|---|---|---|
| Standard NMOS | 53.2 GHz | 60.7 GHz | 64.1 GHz |
| Low-Vth NMOS | 57.0 GHz | 64.2 GHz | 67.1 GHz |
| Standard PMOS | 15.3 GHz | 19.2 GHz | 21.7 GHz |
| Low-Vth PMOS | 20.7 GHz | 24.9 GHz | 27.0 GHz |

The transition frequency $f_T$ of transistors is derived from the unity current gain. $f_T$ is a performance metric of the technology. It gives an intuitive guide to the high-speed amplifier design.

Table 4 shows the $f_T$ of different transistor models. Since $f_T$ is proportional to the overdrive voltage $V_{OV}$ in a first-order equation, the larger $V_{OV}$ results in the higher ft. The result in Table 4 is based on an ideal pre-simulation. All transistors are free of body effect.

Both the low-Vth NMOS/PMOS are adopted for the implementation of DCDL.

### 4.2.4 Inductive Peaking

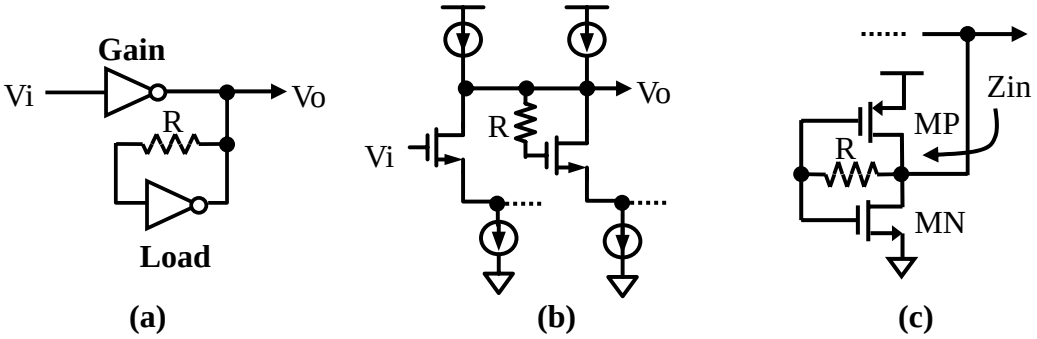



Fig. 4.4: Schematics of (a) the proposed inductive peaking, (b) an inductive-peaking folded amplifier in half circuits, (c) the inductive load.

Fig. 4.4(a) shows the schematic of the proposed inductive. It consists of the input gain stage and the inductive load. The inductive load simulates the frequency-dependant impedance of a passive inductor. The architecture in Fig. 4.4(a) is similar to that in Fig. 4.4 (b), except that 1) the current sources are removed. And, 2) PMOS are active in the signal path.

Since PMOS loads the signal path, the bandwidth of the DCDL is mainly limited by PMOS. Besides, $f_T$ of the low-Vth PMOS is 25-GHz at the 300-mV overdrive in 0.13-um technology. It is therefore critical to achieve the 10-Gb/s operation, which includes the amplification at the pre-amplifier stage.

Fig. 4.4(c) shows the schematic of the inductive load. Define the equivalent $g_m'$ and $g_{ds}'$ as

$$g_m' \equiv g_{m,P} + g_{m,N} \tag{4.4}$$

$$g_{ds}' \equiv g_{ds,P} + g_{ds,N}$$

$$G \equiv R^{-1}$$

where $g_{m,P}$ denotes the gm of PMOS. $g_{m,N}$ denotes the gm of NMOS. The input impedance of the load stage is

$$Z_{in}\big|_{dc} = (g'_{ds} + g'_m)^{-1} \approx g'_m{}^{-1} \tag{4.5}$$

$$Z_{in}\big|_{high\ frequency} = (G + g'_{ds})^{-1} \tag{4.6}$$

In (4.5), the low-frequency input impedance $Z_{in}$ of the inductive load is

$(g_m')^{-1}$ since the local feedback resistor generates the drain-to-gate DC path for the diode-connect MOS. In (4.6), the high-frequency $Z_{in}$ of the inductive load is the resistance R in parallel with the output impedance of the inverter. Because, the equivalent gate-to-source capacitance forms a short-circuit path at high frequency. To introduce the inductive peaking, the impedance $(G + g'_{ds})^{-1}$ at high frequency is designed to be larger than the impedance $(g_m')^{-1}$ at low frequency.

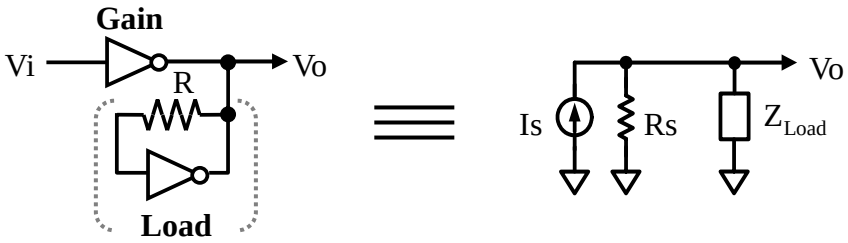



Fig. 4.5: Simplified small-signal model of the proposed inductive peaking

Fig. 4.5 shows the simplified small-signal model to derive the DC gain. The gain stage is modeled by a current source *Is* and a resistor *Rs*. The inductive load is modeled by a simple $Z_{Load}$. To express Is, Rs, and Vo in the equivalent gm and gds, we get

$$Is = g'_{m,Gain} V_i \tag{4.7}$$

$$R_s = (g'_{ds,Gain})^{-1} \tag{4.8}$$

$$Z_{Load}\big|_{dc} = (g'_{ds,Load} + g'_{m,Load})^{-1} \tag{4.9}$$

$$V_o = Is \cdot (R_s \;//\; Z_{Load}) \tag{4.10}$$

$$= g'_{m,Gain} V_i \cdot (g'_{ds,Gain} + g'_{ds,Load} + g'_{m,Load})^{-1}$$

So, the DC voltage gain is

$$A_v(0) = \frac{V_o}{V_i}\bigg|_{dc} = \frac{g'_{m,Gain}}{g'_{ds,Gain} + g'_{ds,Load} + g'_{m,Load}} \approx \frac{g'_{m,Gain}}{g'_{m,Load}} \tag{4.11}$$

As mentioned, PMOS with its transition frequency of 25 GHz limits the operation speed. So, the DC gain of the pre-amplifier cell is designed to be approaching 2 but less than 2.

Return to the delay cell in Fig. 4.3(a). It is composed of 1) *the gain stage*, 2) *the load stage*, and 3) *two inverters in the output path*. The previous two circuits introduce the inductive peaking. As for the two inverters, they are positive-feedback latches in the delay cell.

Although the latches introduce an extra load to the signal paths, there are more significant benefits listed as below. 1) Relate the data and data-bar paths. 2) Introduce hysteresis by adjusting the upward/downward trip-point. And 3) through hysteresis, introduce more delay and also suppress noise.

# 4.3 Critical Design (2): Loop Latency

The second critical design is to meet the loop-latency constraint. For the counter size N = 24, the circuit process time of the entire loop should be less than 2.4ns, or six 2.5-GHz clock cycles. Static CMOS logic gates are not capable of this task. Dynamic logic circuits are fast in operation and economic in power, but these circuits introduce a complex clock distribution network. As a result, the pseudo-NMOS logic gates are adopted. In the **pseudo-NMOS scheme**, logic functions can be optimized into one dimension, which is the NMOS dimension, while the grounded PMOS just acts as a load. The operation time for a specific function is greatly reduced because of the reduced number of stages.

Fig. 4.6 shows the loop latency diagram. It introduces the retiming pipeline stages. The loop has been broken at *the output of DCDL* and *the input of PDs*. A partial retiming scheme is applied from phase detectors to the XOR-Group stage to minimize the number of stages. The output Q<5:7> and Q<0> from phase detectors do the XOR function together with the retimed output Qr<0:4>. The 4 sets of Lead/Lag are asynchronous but behave synchronously as long as the timing is well designed. Fig. 4.7 illustrates the timing of signal flow from phase detectors to the 3-bit encoders.

The first 50-ps latency in Fig. 4.6 results from the setup time of the phase detector. Notice that the global 2.5-GHz clock *Ck* aligns to the clock *P0*. The phase difference between P<n> and P<n+1> is 50 ps. So, the retiming latency from Q0 to Qr0 is 400 ps, and from Q1 to Qr1 is 350 ps … and so forth. As for the latency of the direct-forward Q0 is 0 ps. The total loop latency is 2.45 ns to 2.05 ns, where the

2.45-ns latency comes from the longest Q0-to-Qr0 path. The average loop latency is 2.25 ns, which is less than 2.4 ns of the loop-latency constraint.

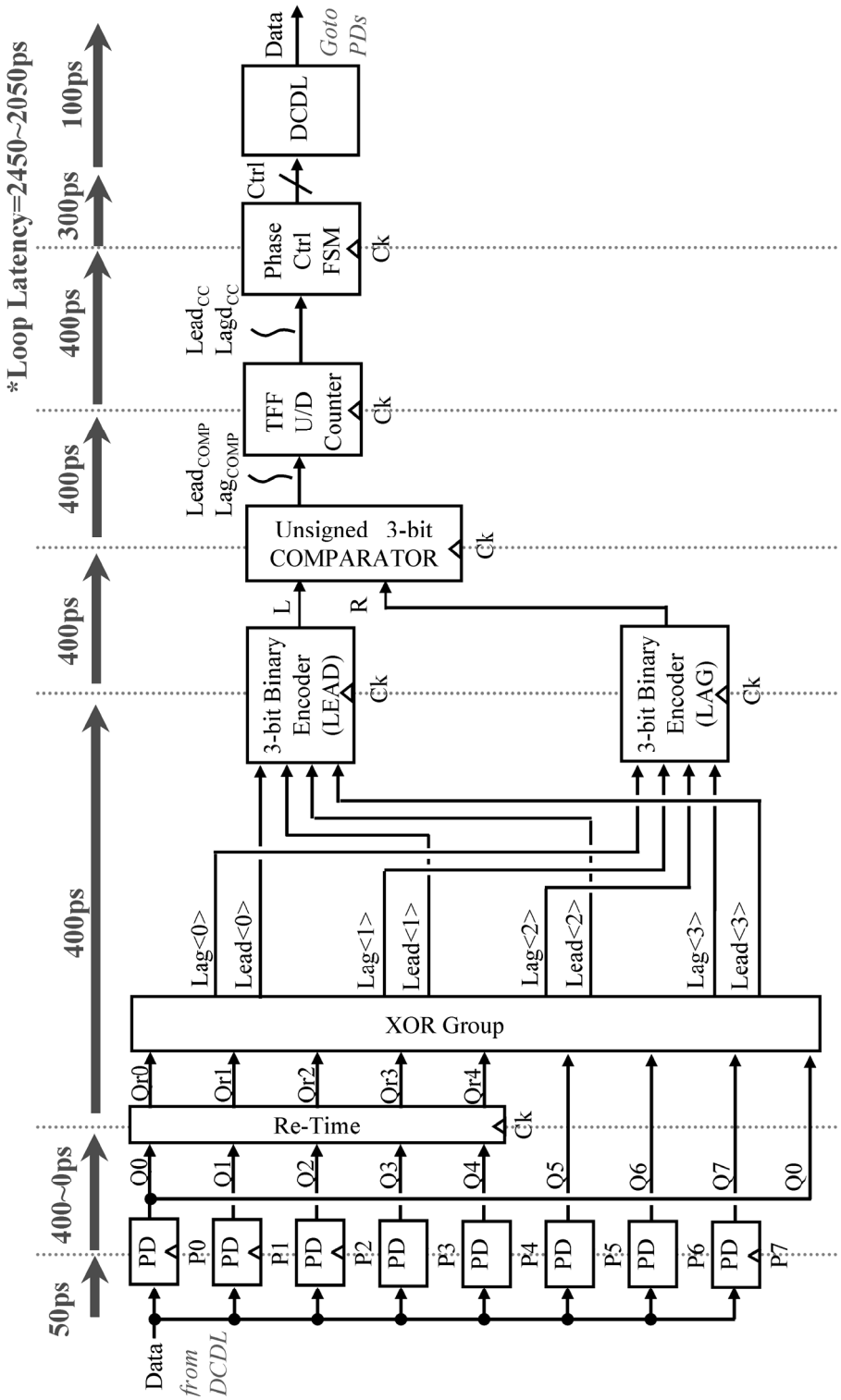



Fig. 4.6: The loop-latency timing

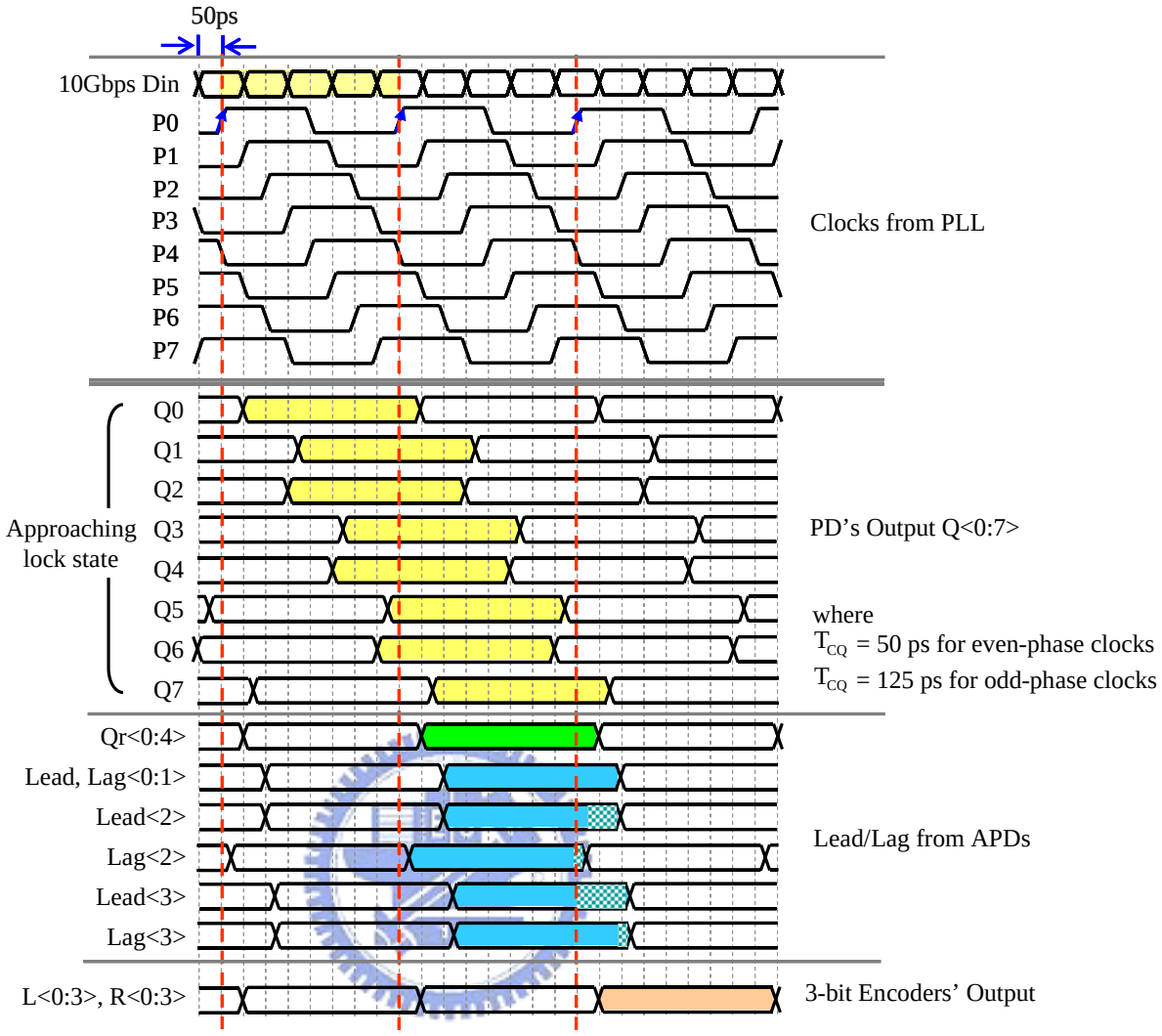



Fig. 4.7: The timing diagram: *from PDs to the 3-bit encoders*

# 4.4 Building Blocks

## 4.4.1 Alexander Phase Detector

Alexander phase detector [29], or APD, was published on Electronics Letters in 1975. Because of the high-gain characteristics in its input-to-out transfer plot, it is also known as a *bang-bang* PD or a binary quantized PD. Fig. 4.8(a) shows the conventional APD architecture. Fig. 4.8(b) shows the timing diagram. An assumption $T_{CQ} = 0$ has been made.

In Fig. 4.8(b) and (c), the input *Di* is sampled at both the rising/falling edges of

the clock. Node *a* is sampled at the first rising edge. Node *b* is sampled at the falling edge. Node *c* is sampled at the last rising edge. So, nodes *a*, *b*, and *c* are all sampled and retimed at the rising edges. It ensures the synchronous operations of the XOR gates.

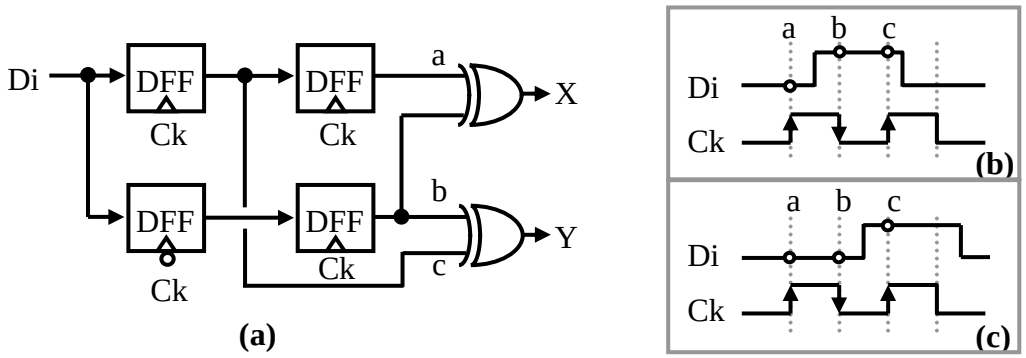



Fig. 4.8: Conventional APD (a) schematic (b) data leads (c) data lags

Consider the clock falling edge as a reference edge. Fig. 4.8(b) is the case that the data edge leads the reference edge. Fig. 4.8(c) implies the data edge lags. In the timing diagram, the data Lead/Lag phenomenon can be observed by the logic operations of a, b, and c. If $(X \cdot \overline{Y})$ is true, the data edge leads the clock falling edge. If $(Y \cdot \overline{X})$ is true, the data edge lags behind the clock falling edge. Fig. 4.8 illustrates the full-rate APD, where the clock rate equals to the data rate. The clock falling edge also samples the data. Conceptually the APD applies an oversampling scheme to recover the NRZ data, where the oversampling ratio is 2.

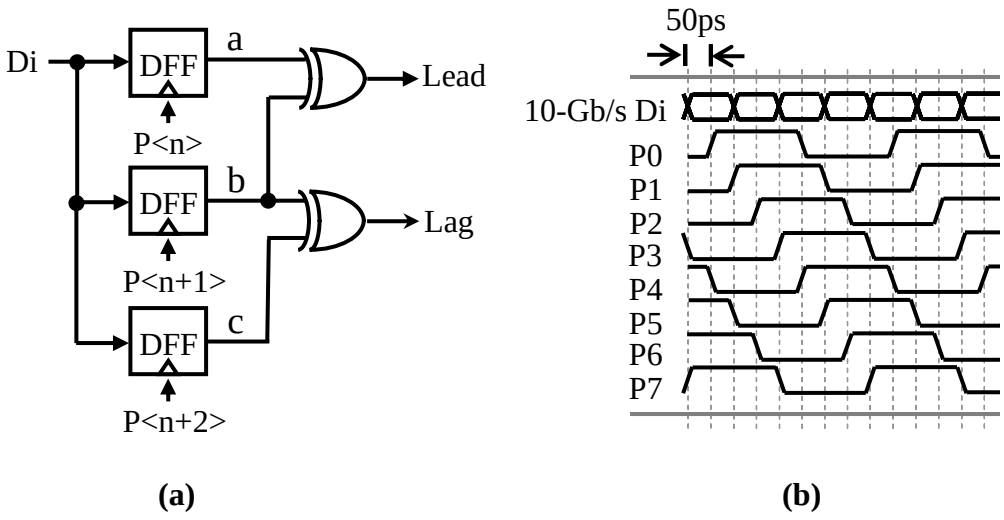



Fig. 4.9: Proposed APD (a) schematic (b) timing diagram

Fig. 4.9(a) shows the proposed quarter-rate APD, where n = 0, 2, 4, 6. It applies the 2.5-GHz 8-phase clocks shown in Fig. 4.9(b). The Lead/Lag information can be derived from the nodes a, b, and c as previously described. In Fig. 4.8(a), $(X \cdot \overline{Y})$ validates for data edge leading and $(Y \cdot \overline{X})$ validates for data edge lagging. But for the proposed implementation, it simply regards X as Lead, and Y as Lag to reduce the process time. This simplification introduces two false events shown in Table 5. Later on these two events are corrected by the majority-vote scheme in the confidence counter.

Table 5: The truth table of the proposed APD

| a | b | c | Lead | Lag |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 |

## 4.4.2 Confidence Counter

Fig. 4.10 shows the confidence counter architecture. It consists of 1) a 3-bit unsigned encoder, 2) a comparator, and 3) an accumulator.

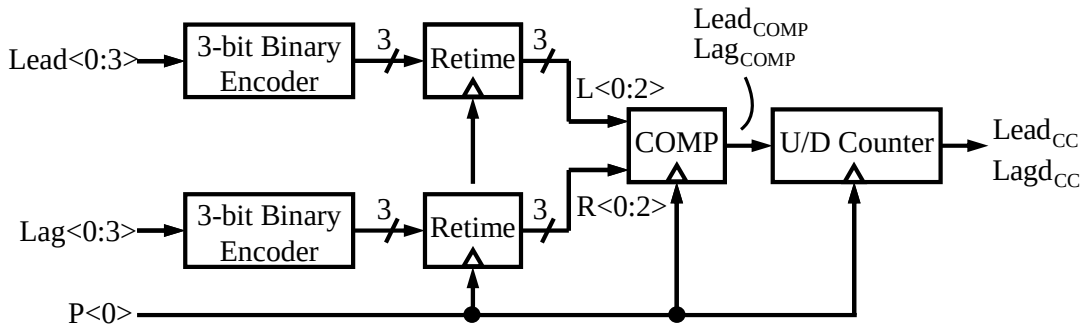



Fig. 4.10: Confidence counter architecture

## ■ The 3-bit Encoders

The encoders derive the counts of Lead<0:3> and Lag<0:3>. For the circuit implementation, it counts the zeros of the input-bars. The result is encoded into 3-bit unsigned integer O<0:2> shown in Table 6.

Table 6: The truth table of the encoder

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I3b | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| I2b | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| I1b | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| I0b | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| O2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| O1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| O0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

Denote I3b as *a*, I2b as *b*, I1b as *c*, and I0b as *d*. The output *O* can be expressed as below

$$O2 = \overline{a+b+c+d} \tag{4.12}$$

$$O1 = (\bar{a}\cdot b + a\cdot\bar{b})(\bar{c}+\bar{d}) + \bar{a}\cdot\bar{b}(c+d) + a\cdot b\cdot\bar{c}\cdot\bar{d}$$

$$O0 = (a\oplus b)\oplus(c\oplus d)$$

## ■ The Comparator

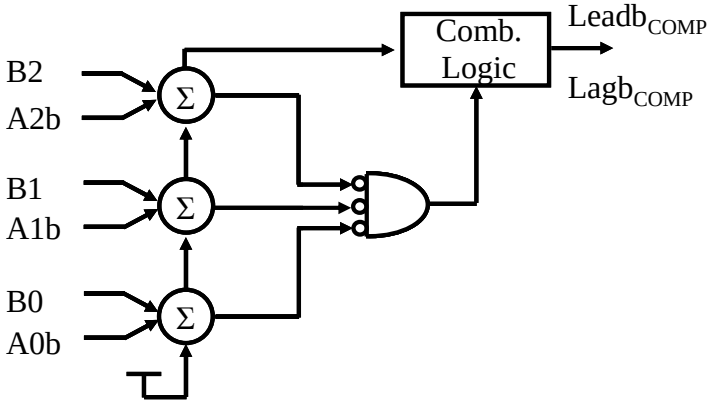



Fig. 4.11: The 3-bit binary comparator

Fig. 4.11 shows the 3-bit comparator implemented with a ripple adder. The comparison result is derived from (B - A). The circuit implementation adopts the carry-look-ahead adder instead of the ripple adder, so the long carry chain is avoided.

The look-ahead carries can be expressed as

$$C_1 = P_0 \cdot C_{in} + G_0 = P_0 + G_0 \tag{4.13}$$

$$C_2 = P_1 \cdot (P_0 + G_0) + G_1$$

$$C_3 = P_2 \cdot P_1 \cdot (P_0 + G_0) + P_2 \cdot G_1 + G_2$$

where *P* indicates the propagation of the carry. *G* indicates the generation of the carry. The pseudo-NMOS circuits for the carry generation are given in Fig. 4.12.

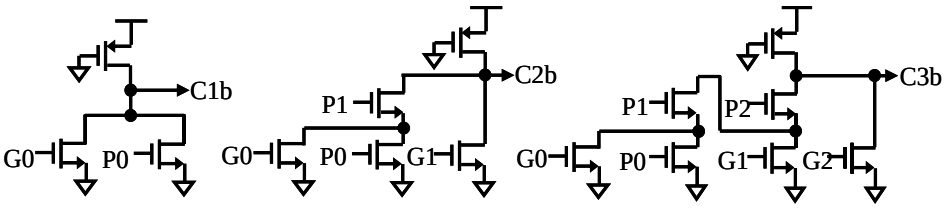



Fig. 4.12: Circuits of look-ahead carries

■ **The Accumulator**

The last stage of confidence counter is an accumulator. It is an up-down counter as shown in Fig. 4.13. When either $\text{Leadb}_{\text{COMP}}$ or $\text{Lagb}_{\text{COMP}}$ from the comparator validates, the accumulator starts its counting. The validation of $\text{Leadb}_{\text{COMP}}$ results in upward counting, while that of $\text{Lagb}_{\text{COMP}}$ results in the downward counting.

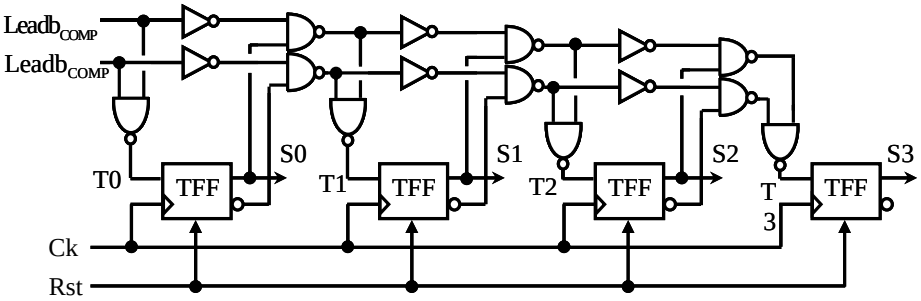



Fig. 4.13: The main counter in the accumulator

The critical path in Fig. 4.13 is the *T's logic*. It can be optimized by the equations below

$$T_0 = A + B = \overline{\overline{A} \cdot \overline{B}} \tag{4.14}$$

$$T_1 = AS_0 + B\overline{S_0} = \overline{\overline{AS_0} \cdot \overline{B\overline{S_0}}}$$
$$= \overline{(\overline{A} + \overline{S_0}) \cdot (\overline{B} + S_0)}$$

$$T_2 = AS_0S_1 + B\overline{S_0}\,\overline{S_1} = \overline{\overline{AS_0S_1} \cdot \overline{B\overline{S_0}\,\overline{S_1}}}$$
$$= \overline{(\overline{A} + \overline{S_0} + \overline{S_1}) \cdot (\overline{B} + S_0 + S_1)}$$

$$T_3 = \overline{(\overline{A} + \overline{S_0} + \overline{S_1} + \overline{S_2}) \cdot (\overline{B} + S_0 + S_1 + S_2)}$$

where $\text{Leadb}_{\text{COMP}} / \text{Lagb}_{\text{COMP}}$ is denoted as $\overline{A} / \overline{B}$. Based on (4.14), Fig. 4.14 realizes the T's logic in the pseudo-NMOS scheme. The process time is greatly reduced. It approximates one-gate delay.

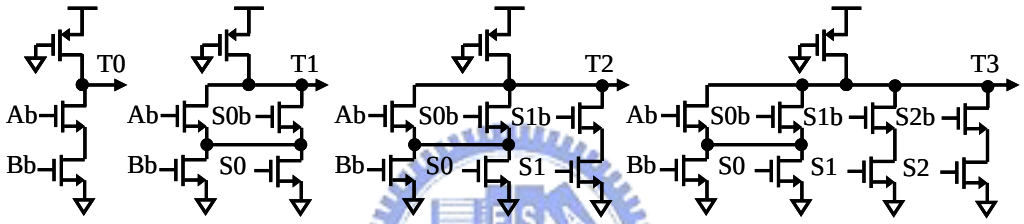



Fig. 4.14: T's logic in pseudo-NMOS

Fig. 4.15 shows the state transition diagram of the accumulator. The upward counting is denoted as **U**. The downward counting is denoted as **D**. And, **N** stands for the case that both U and D are false. The path (1) generates the $\text{Lag}_{\text{CC}}$. The path (2) generates $\text{Lead}_{\text{CC}}$.

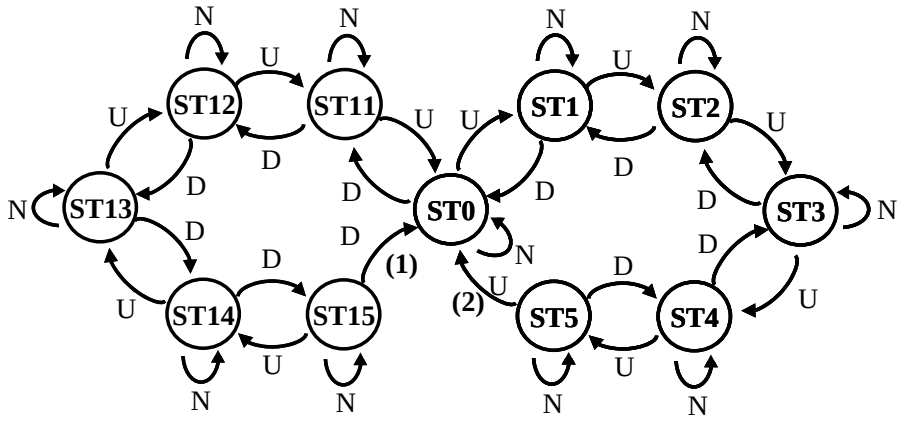



Fig. 4.15: The state transition diagram of the accumulator

Table 7 shows the state name and its counting value. If the accumulator overflows, outputs $Lead_{CC}$ / $Lag_{CC}$ will be validated. The counter will be reset to its initial state 'ST0' immediately.

Table 7: The state table of the accumulator

| State Name | ST5 | ST4 | ST3 | ST2 | ST1 | ST0 | ST11 | ST12 | ST13 | ST14 | ST15 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Counting Value | 5 | 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 | -4 | -5 |

Implementing the dynamic reset function is tricky. In Fig. 4.13, the main counter employs the resetable T Flip-Flops. The input $Leadb_{COMP}$ / $Lagb_{COMP}$ and the current state are monitored all the time. Once the overflow condition is hit, the default setting[7] is loaded to the TFFs at the next clock cycle.

### 4.4.3 Finite State Machine

FSM consists of an up-down counter and combinational logic circuits. In Fig. 4.16, the 5-bit counter is implemented with TFFs just like the 4-bit counter in confidence counter block. Comparing to the dynamic reset of the confidence counter, the reset function here is static and is for the initialization of DCDL. It adds the delay of half the tuning range to the 10-Gb/s input data.

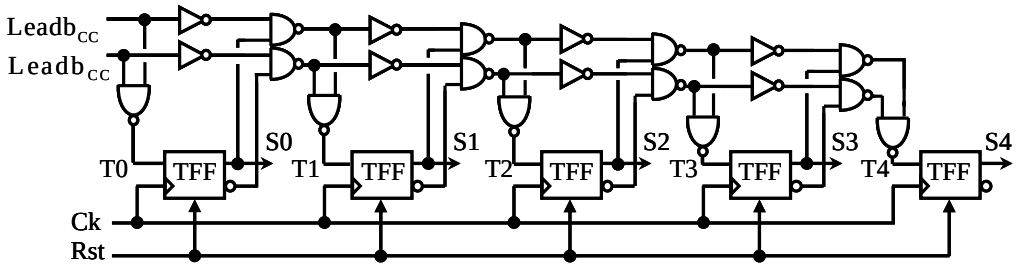



Fig. 4.16: The 5-bit counter in FSM

The coarse/fine control codes are developed from the counting value of the main counter. S<2:4> determines the coarse control C<0:7>, and S<0:2> determines the fine control F<0:3>. Table 8 shows the truth tables for coarse/fine control codes. For S<2:4> = [ 1, 1, 1 ], it is an illegal state.

[7] The default setting can be any counting value in binary format. It is 'all zero' in this implementation.

Table 8: The truth tables of FSM (a) coarse tuning function (b) fine tuning function

| S4 | S3 | S2 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | illegal |

**(a)**

| S2 | S1 | S0 | F3 | F2 | F1 | F0 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

**(b)**

Table 9: The complete truth tables of FSM

| S4 | S3 | S2 | S1 | S0 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 | F3 | F2 | F1 | F0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | Initial State |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |

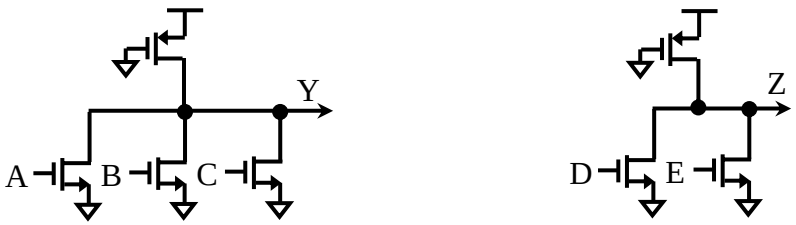
Table 9 shows 28 sets of control codes. In other words, there are 27 phase intervals available for the tuning. The initial state S<0:4> = [ 0, 1, 1, 1, 0 ] adds the delay of half the tuning rage to the data.

The generation of coarse control *C<0:7>* is critical in FSM. Even though they come from the synchronous S<2:4>, the combinational logic is asynchronous and may introduce timing difference to C<0:7>. Consider a timing difference of 1 UI for example. The 4-to-1 multiplexers in DCDL fail to select the correct data path, so one bit of the full-ratet data is missing.

To minimize the timing difference between C<0:7>, the coarse control adopts the *two-stage NOR scheme* in pseudo-NMOS. The equations are given as below

$$\begin{aligned}
e_0 &= \overline{S_4}\cdot\overline{S_3}\cdot\overline{S_2} = \overline{S_4+S_3+S_2} \;;\; \overline{C_0} = \overline{0+e_0} \\
e_1 &= \overline{S_4}\cdot\overline{S_3}\cdot S_2 = \overline{S_4+S_3+\overline{S_2}} \;;\; \overline{C_1} = \overline{e_0+e_1} \\
e_2 &= \overline{S_4}\cdot S_3\cdot\overline{S_2} = \overline{S_4+\overline{S_3}+S_2} \;;\; \overline{C_2} = \overline{e_1+e_2} \\
e_3 &= \overline{S_4}\cdot S_3\cdot S_2 = \overline{S_4+\overline{S_3}+\overline{S_2}} \;;\; \overline{C_3} = \overline{e_2+e_3} \\
e_4 &= S_4\cdot\overline{S_3}\cdot\overline{S_2} = \overline{\overline{S_4}+S_3+S_2} \;;\; \overline{C_4} = \overline{e_3+e_4} \\
e_5 &= S_4\cdot\overline{S_3}\cdot S_2 = \overline{\overline{S_4}+S_3+\overline{S_2}} \;;\; \overline{C_5} = \overline{e_4+e_5} \\
e_6 &= S_4\cdot S_3\cdot\overline{S_2} = \overline{\overline{S_4}+\overline{S_3}+S_2} \;;\; \overline{C_6} = \overline{e_5+e_6} \\
e_7 &= S_4\cdot S_3\cdot S_2 = \overline{\overline{S_4}+\overline{S_3}+\overline{S_2}} \;;\; \overline{C_7} = \overline{e_6+0}
\end{aligned} \tag{4.15}$$

The logic gates of NOR3 and NOR2 are shown in Fig. 4.17. By adopting (4.15), the timing difference is eliminated. But, it requires two stages to finish the logic. An extra stage delay is introduced to the loop.




**(a)** **(b)**

Fig. 4.17: NOR gates for coarse control (a) NOR3 (b) NOR2

The fine control *F<0:3>* determines the interpolation. It is relatively robust, because 1) the summation of weighted proportions is inherently linear, and 2) the interpolation is ensured by the thermometer coding style. The equations of F<0:3> are given

$$F_3 = S_2 \tag{4.16}$$

$$F_2 = S_2 \oplus (S_1 \cdot S_0)$$

$$F_1 = S_2 \oplus S_1$$

$$F_0 = \overline{S_2} \oplus (\overline{S_1} \cdot \overline{S_0})$$

Fig. 4.18 shows the circuit implementation of $Y = A \oplus (B \cdot C)$.

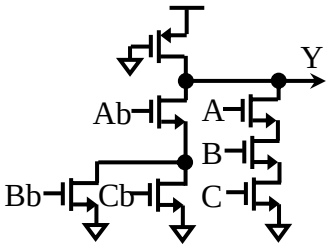



Fig. 4.18: The circuit of $Y = A \oplus (B \cdot C)$ for the fine control

### 4.4.4 Logic Components

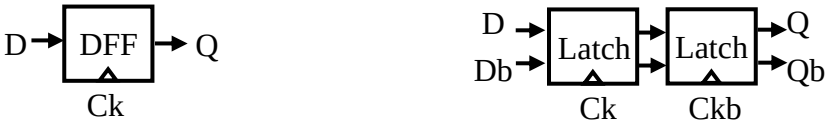
This section shows the circuits of some basic logic components which build up the system. In the pseudo-NMOS scheme, the process time of each functional block is one-gate delay.

■ **XOR/XNOR**


**(a)** **(b)**

Fig. 4.19: (a) XOR (b) XNOR

■ **D Flip-Flop**




**(a)** **(b)**

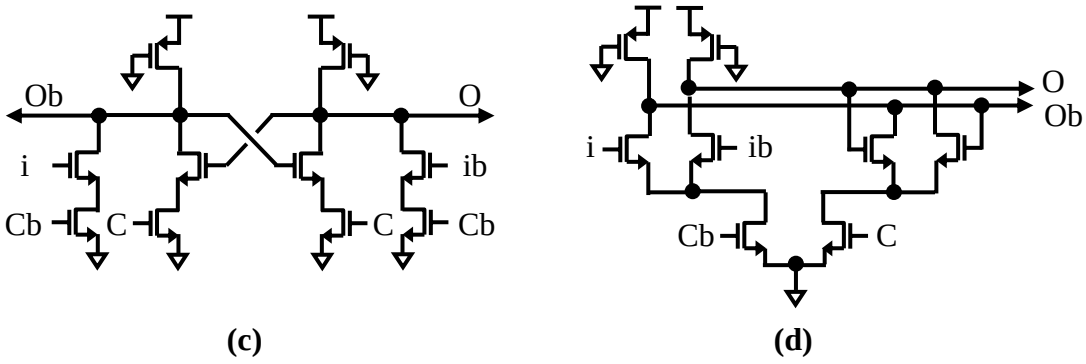



Fig. 4.20: Schematics of (a) a DFF, (b) master & slave latches, (c) a latch, (d) the source-coupling latch.

■ **T Flip-Flop**

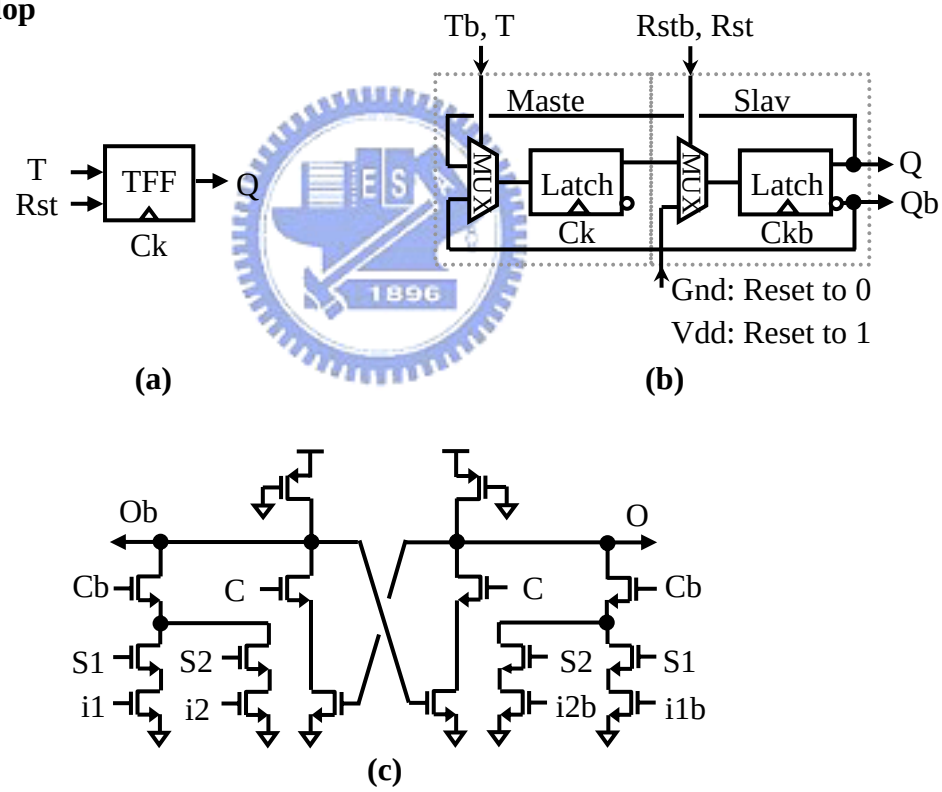



Fig. 4.21: Schematics of (a) a resetable TFF, (b) master & slave stages of the TFF, (c) the circuit of MUX & latch.

# Chapter 5

# A 10-Gb/s Transceiver Test Chip

## 5.1 Architecture

The test chip realizes a 10-Gb/s transceiver, which consists of a CDR and a transmitter. Fig. 5.1 shows the full-chip architecture. The core circuit is the proposed Deskew CDR. It receives the full-rate data from the tester and recovers it into four quarter-rate data D<0:3>, which synchronize to the global clock P<0>. The parallel data then go to the transmitter, which is composed of the 4-to-1 serializer and the full-rate output buffer, and then the serialized full-rate data go back to the tester.

Due to the lack of on-chip channels in this chip, the channel-induced jitter and the degradation on signal amplitude is modeled by the CDR's input from the tester. Functions of *jitter injection, data driving range, and crossing point adjustment, etc.* are available by the tester.

This chip is realized with high flexibility on measurements, based on two facts. First, the full-rate differential data and clock come from the tester synchronously. Second, the on-chip quarter-rate clocks are derived from the full-rate clock. The performance of the chip can thus be fully measured. For example, the original

targeting data rate is 10 Gb/s, but it can be measured at 12 Gb/s for the up-grade case or at 8 Gb/s for the down-grade case.

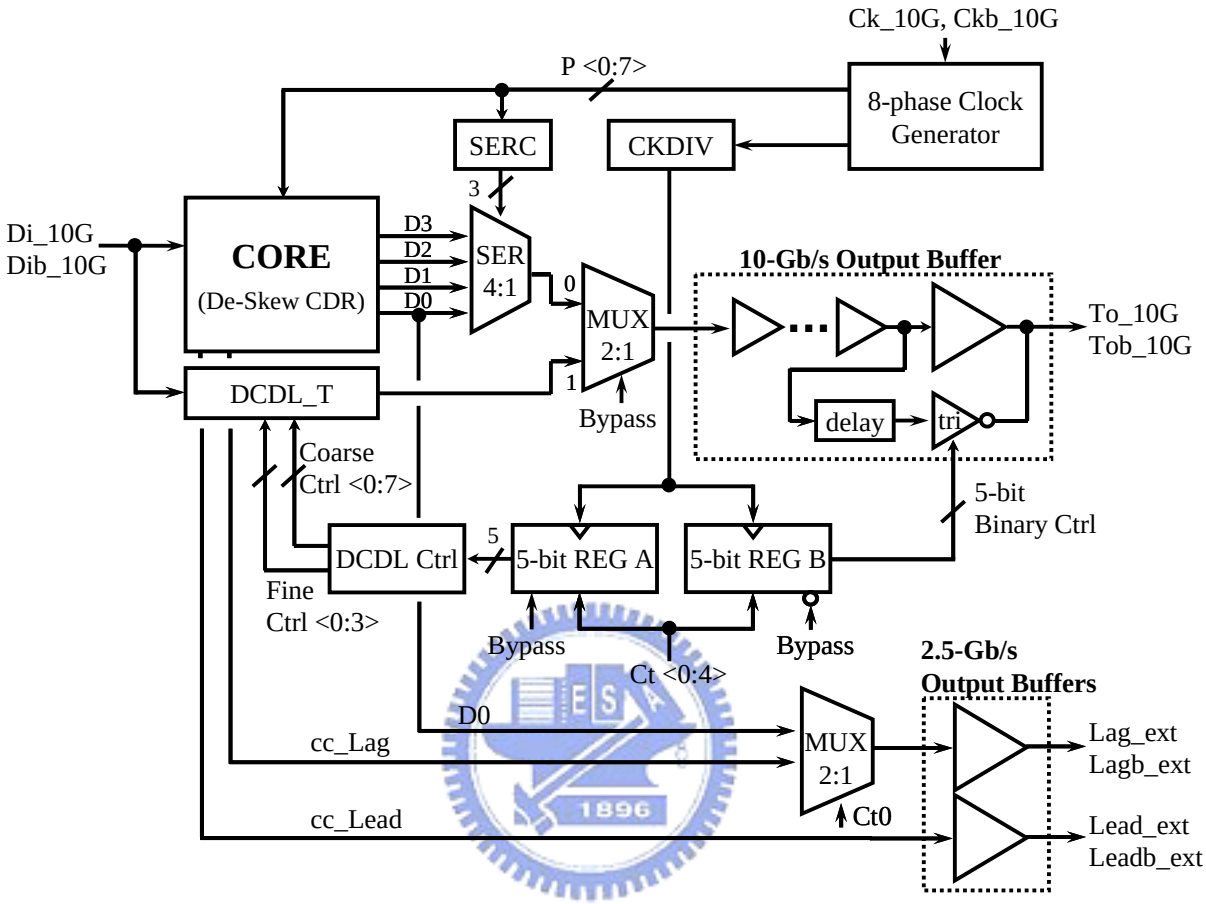



Fig. 5.1: The full-chip architecture

# 5.2 Design for Test

Besides the previous description on controlled data input and the measurement flexibility, this section describes the considerations for the test, such as phase resolution measurements, pads/buffers sharing, etc, as well as the circuit operation in different modes.

## 5.2.1 Phase Resolution Measurement

An additional DCDL block, denoted as DCDL_T, is introduced to the chip. There are mainly two measurements to be made. First is to understand the phase resolution of DCDL. Second is to see how the DCDL-induced jitter affects the output

eye diagram. And from the eye diagram of different tapping location, the induced jitter from stage to stage is observed. The maximum number of stages can be further analyzed and determined by the measurement results.

### 5.2.2 Output Buffer Driving Capability

The driving capability of the full-rate output buffer is controllable through the variable amount of the pre-emphasis. The delayed negative-feedback driving strength of the tri-state buffers constitutes the pre-emphasis mechanism. For a large capacitive load, it requires more strength but at the cost of the eye closure in the magnitude axis. As for a small capacitive load, less strength is able to deal with. A capacitive load of 1 pF for each probe pad is considered as the typical case in this work.

The negative-feedback strength depends on how many tri-state buffers are activated, and therefore it is controllable.

### 5.2.3 Pads/Buffers Sharing

The control mechanism of DCDL_T and the driving strength of the output buffer both require 5 bits in binary format. To save the area cost, these 10 bits share the 5 control pads, denoted as Ct<0:4>.

Two sets of 5-bit registers are used for the pads sharing. They operate in the low-frequency domain. The low-frequency clock comes from the global 2.5-GHz clock divided by 512, so as to eliminate the disturbance to the internal circuits.

The 5-bit data Ct<0:4> is loaded continuously to either Register A or Register B. The *bypass* signal determines which to be loaded.

The 10-Gb/s output buffer is the only way out for 10-Gb/s data for sure. Two inputs, multiplexed by the bypass signal, share the buffer. And, a 2.5-Gb/s output buffer is also shared by a multiplexer, which is controlled by the signal Ct<0>.

### 5.2.4 Bypass-Mode Operation

The first thing to test is the bypass-mode operation. Fig. 5.2 shows the active blocks and the signal flow in bypass mode. The full-rate data passes through DCDL_T, 2-to-1 full-rate multiplexer, and finally output from the buffer. This bypass mode makes sure that the full-rate data path behaves well.

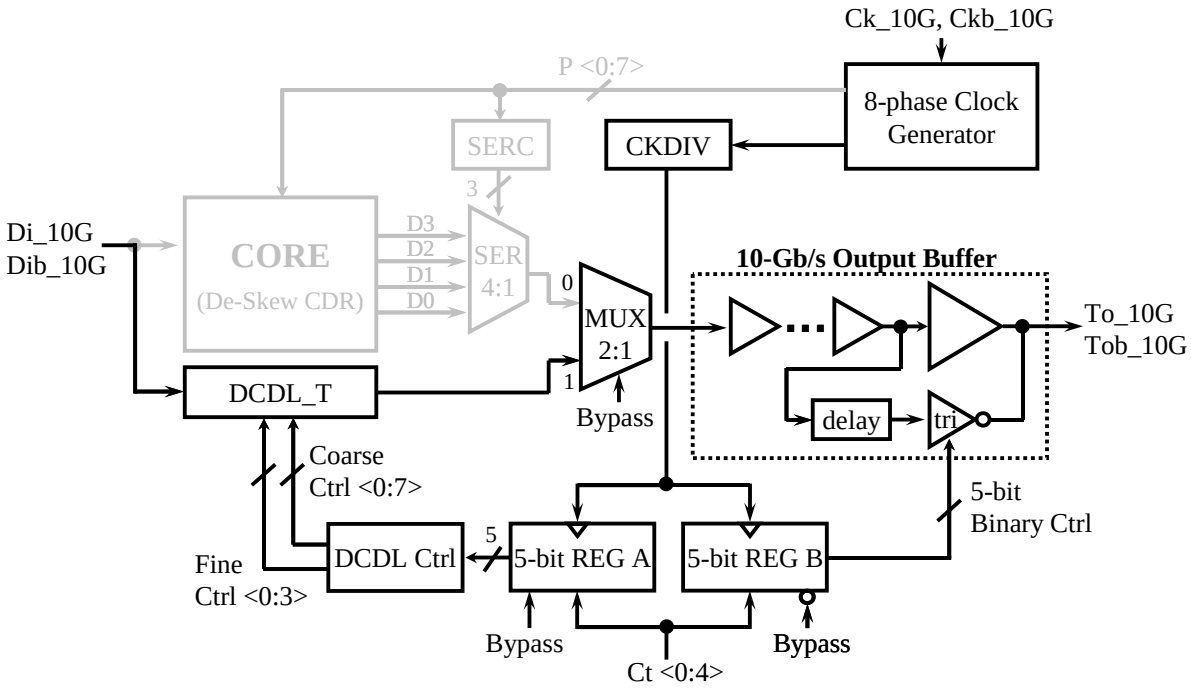



Fig. 5.2: Bypass-mode operation

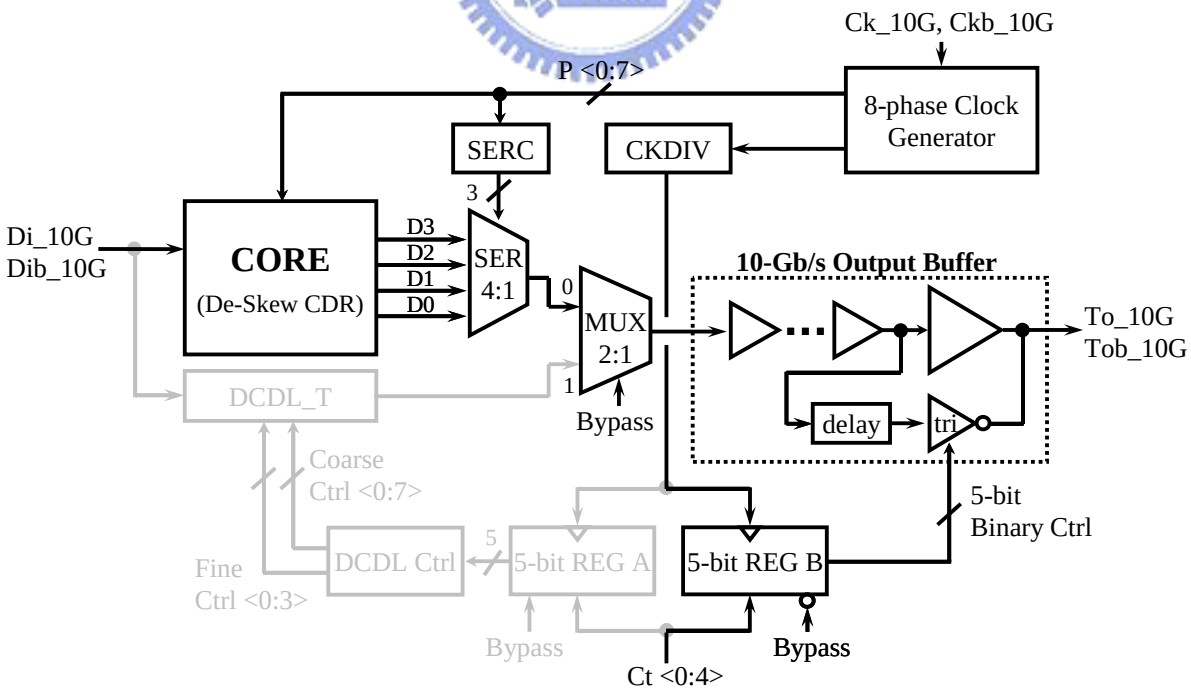



Fig. 5.3: Nominal operation

The phase resolution of DCDL_T is measured in bypass mode. Before the measurement, configuring two settings is required. First, configure the output buffer setting. Set the registers B at bypass = 0 . After that, set bypass = 1 and enter the bypass mode. Configure the setting of registers A for DCDL_T. The phase resolution of each step can be measured by stepping the setting of registers A.

### 5.2.5 Nominal Operation

The chip is in nominal mode when bypass signal is false. Fig. 5.3 shows the corresponding active blocks and the signal flow. The full-rate data is recovered and de-serialized in Deskew CDR. It is serialized again in the 4-to-1 serializer. A 2-to-1 multiplexer selects the full-rate data at bypass = 0 . Finally the output buffer drives the full-rate data back to the tester. And the driving capability of the output buffer is configured by registers B.

### 5.2.6 Debug-Mode Operation

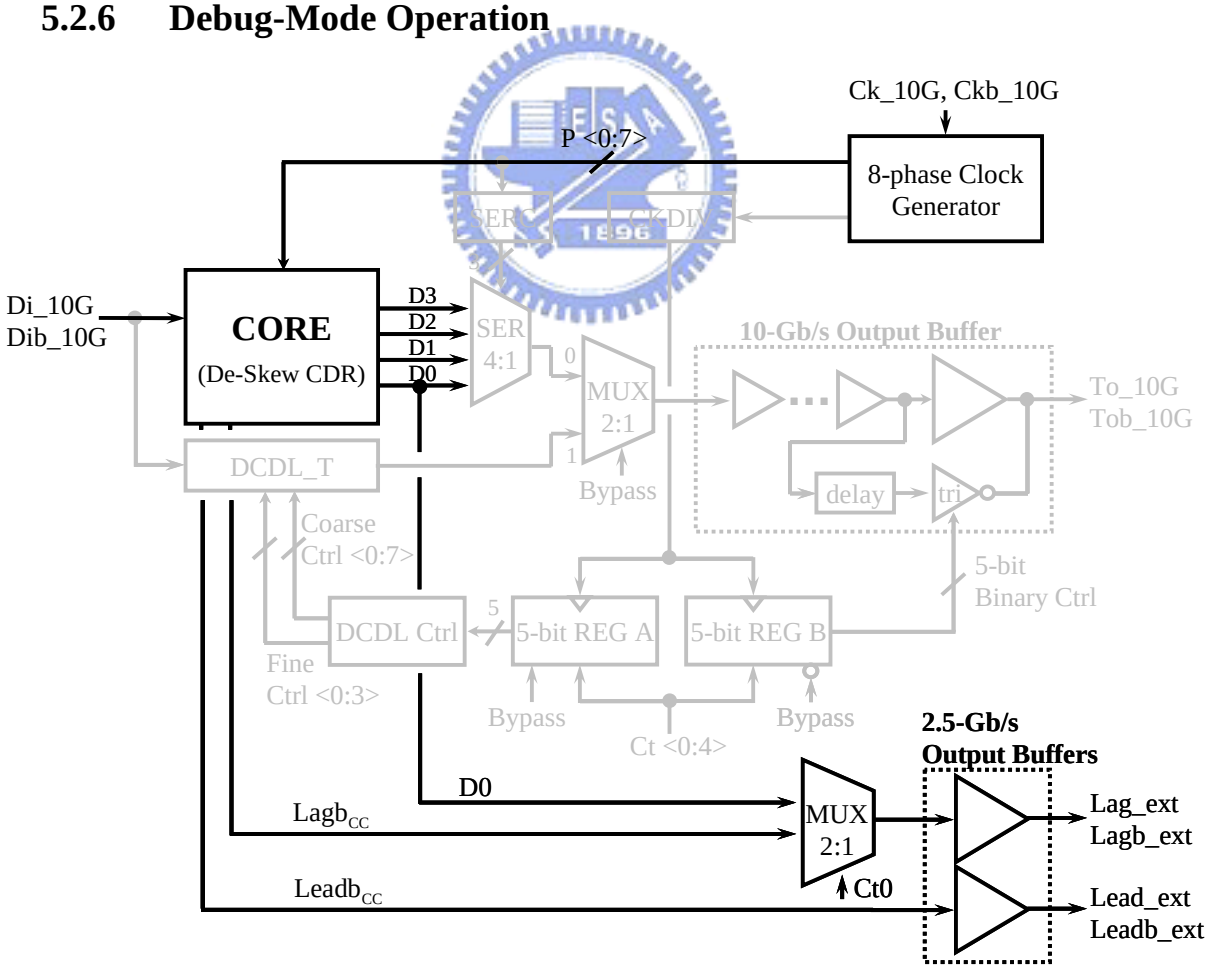



Fig. 5.4: Debug-mode operation

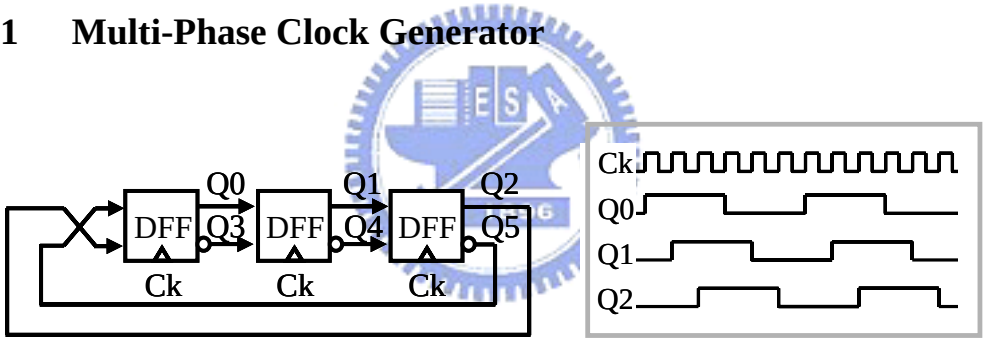
Tracking and locking behaviors of the CDR can be observed from the internal $\text{Lead}_{CC}$ / $\text{Lag}_{CC}$. Fig. 5.4 shows that two 2.5-Gb/s buffers drive these two signals to the oscilloscope in the outer world, and one of the four parallel data from the CDR can be multiplexed to the buffer.

The debug mode is not controlled by any signal, and thus it can accompany the nominal mode. Since the two 2.5-Gb/s output buffers use their own pair of Vdd/Gnd, the supplies can be removed at any time. In that case, the chip is naturally out of debug mode.

# 5.3 Building Blocks

This section depicts three blocks relating to 10-GHz or 10-Gb/s. It involves a multi-phase clock generator, a serializer, and the output buffer.

## 5.3.1 Multi-Phase Clock Generator




**(a)** **(b)**

Fig. 5.5: A 3-stage Johnson counter (a) architecture (b) timing diagram

Fig. 5.5(a) shows a 3-stage **Johnson counter**. The phase difference between Q0 and Q1 is one Ck period. Consider the timing diagram in Fig. 5.5(b), the logic ‘1’ propagates from Q0, through Q1, and then to Q2. Q0 falls to zero at the 4th clock edge because the first DFF’s input-bar receives the positive Q2. And then the logic ‘0’ starts to propagate as the previous description on the logic ‘1’. The propagation of logic ‘1’ takes 3 Ck periods and that of logic ‘0’ takes 3 Ck periods, so the data rate of Q is the clock rate divided by 6 for the 3-stage Johnson counter.

Another approach to explain the data rate of Q is the data flow in the loop. Q0 propagates to Q1, to Q2 … , to Q5, and finally back the Q0. It takes 6 Ck periods to complete the loop. The data rate of Q0 is therefore the clock rate divided by 6.

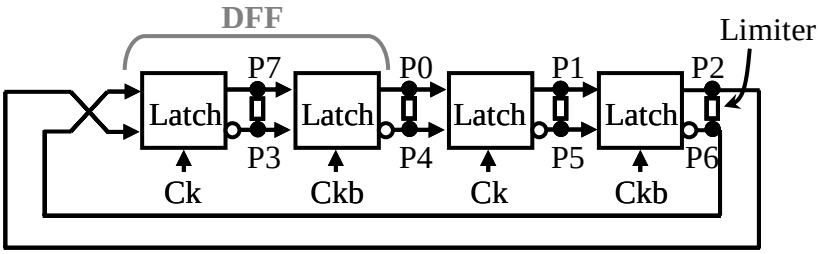



Fig. 5.6: The multi-phase clock generator

The proposed clock divider consists of 4 latches shown in Fig. 5.6. It is a 2-stage Johnson counter since the four latches compose two DFFs. The input clock rate is 10 GHz, and the developed clock rate is divided-by-4 based on the concept of a Johnson counter. To achieve the high-speed operation, limiters for the swing limitation are introduced to the differential signaling.

Even-phase clocks are outputs of DFFs. The phase difference between each even-phase clock is one 100-ps clock period, while the odd-phase clocks are tapped from the master latches of DFFs.

Because the latch circuit is differential and fully symmetric illustrated in Fig. 4.20(d) previously, the phase difference of P<0:7> should be the same. However, the multi-phase outputs from the latch circuits still suffer from the unbalanced duty cycle due to the mismatch between PMOS pulling strength and NMOS sinking strength. To balance the duty cycle, one can adjust the common-mode level of the 10-GHz Ck.

### 5.3.2 Serializer

Fig. 5.7 shows the 4-to-1 serializer circuit. It consists of three 2-to-1 multiplexers in cascaded two stages. The multiplexer at the output stage employs a 5-GHz clock and generates the full-rate serialized data. The multiplexers at the first stage employ two quadrature-phase clocks, and operate in 2.5-GHz domain.

All input data are retimed with buffered clock, which aligns to the global clock P<0>. A DFF implements the 1-cycle delay block, while a DFF with an extra latch implements the 1.5-cycle delay block. The extra latch retimes the data to the clock falling edge, so the 1.5-cycle delay is derived.

It is the conventional way to divide the fastest clock by 2 at the beginning and then generate the lower rate clocks sequentially. In this work, all the serializer clocks are developed from the global clocks P<0:7>.

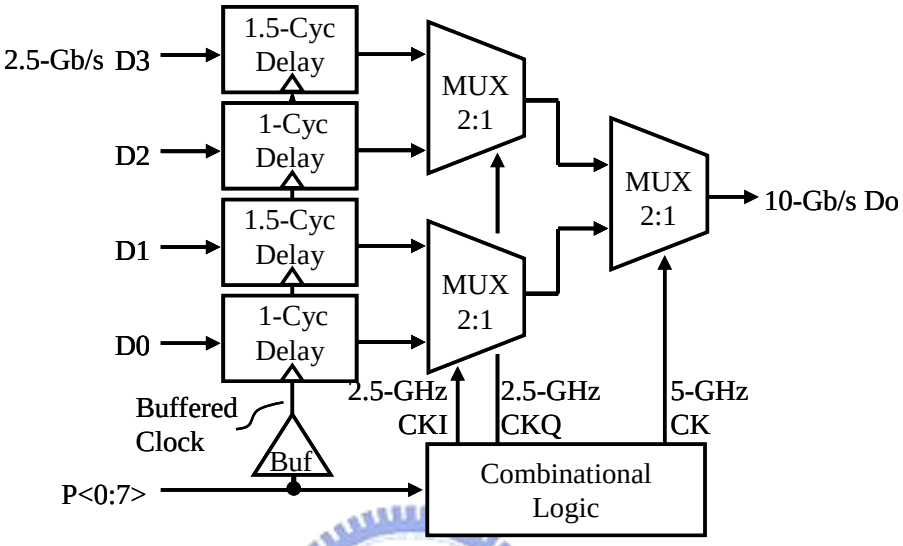



Fig. 5.7: The 10-Gb/s 4-to-1 Serializer

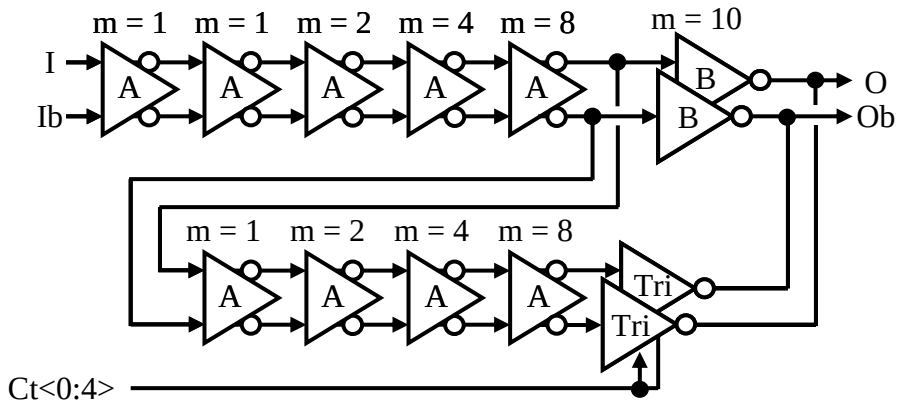
### 5.3.3 Output Buffer

Fig. 5.8(a) shows the digital-circuit implementation of the 10-Gb/s output buffer. It adopts the conventional one-bit delay scheme. The schematic of 'A cell' is the same as the delay cell in Fig. 4.3(a). A cells cascade as tapered buffers, and drive the 'B cell', which is a static CMOS inverter, at the final output stage.

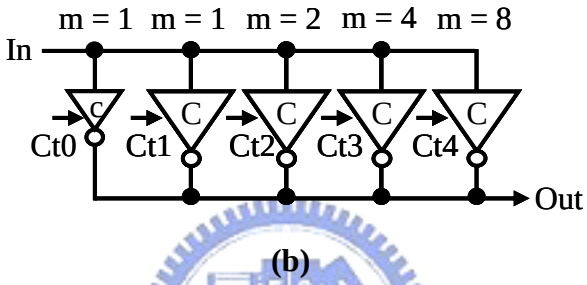
In the bottom of the figure, the delayed-forward loop implements the FIR filter. The output signal of the forward loop is delayed and with negative polarity to the main output of B cell.

The 'Tri' block is composed of the binary-controlled tri-state buffers shown in Fig. 5.8(b), where the 'c cell' is half size of the capital 'C cell'. The amount of the pre-emphasis is controlled by Ct<0:4>. The more turned-on tri-states buffers introduce the larger amount of over-shooting to the output signal in time domain.

Besides, two 50-ohm poly resistors (not shown) are in series and connect the differential output nodes for the impedance matching.




**(a)**




**(b)**

Fig. 5.8: The 10-Gb/s output buffer (a) architecture (b) binary-control tri-state buffers

# Chapter 6

# Layout & Post Simulations

## 6.1 Layout

### 6.1.1 Layout Guidelines for High-Speed Circuits

The layout plan and the block arrangement are critical to the high-speed operation. Based on the iterative process between the layout and the post simulation, the layout guidelines for high-speed circuits are derived. They are listed in the order of benefits to the high-speed operation.

**1) Source Coupling**

It is the most important guideline to the proposed differential large-swing digital circuits. The sources of differential input transistors should be placed together even at the lack of current sources. These sources are connected by the diffusion layer, but contacts sharing are not recommended.

The source coupling layout relates the differential signals. It can be observed in the post simulation with R-C-CC extraction, even though the phenomenon is not

modeled by the circuits in pre-simulation.

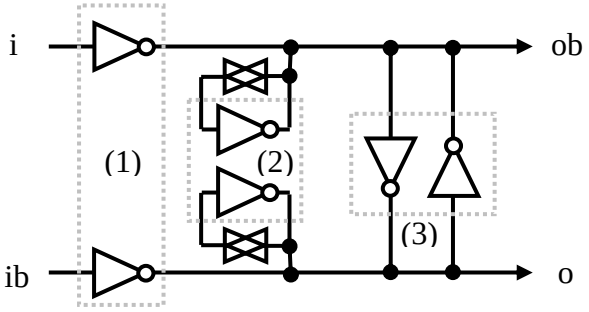



Fig. 6.1: Source-coupling pairs in the delay cell

Fig. 6.1 shows the delay cell of DCDL. Three inverter pairs adopt the source coupling scheme. It improves the 10-Gb/s differential signaling in the delay cell. For the 2.5-GHz domain, the differential signaling also benefits from the source coupling scheme. Fig. 6.2 shows the latch with 3 pairs of NMOS in the source coupling scheme.

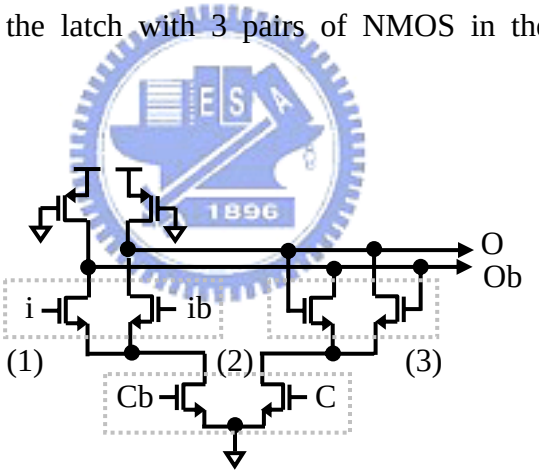



Fig. 6.2: Source-coupling pairs in the latch

**2) Reducing Capacitive Load**

The drain area sharing between two transistors reduces the drain-to-bulk capacitance. Both *the drain area sharing* and *contacts sharing* are recommended.

A capacitive load can be partitioned into three components, 1) the area overlap capacitance $C_a$, 2) the coupling capacitance $C_c$, and 3) the fringing capacitance $C_f$. In most cases, the area overlap capacitance $C_a$ dominates the high-speed performance.

Especially it is observed when the drain's output in the *(N+1)th* metal layer runs across the power line in the *Nth* metal layer. It is recommended to use the *(N+2)th* metal layer and the *(N+2)th* metal layer for the overlapping.

**3) Common-Centroid Geometry**

The common-centroid scheme is a general layout guideline. It mainly solves the process variation across the wafer, but it doesn't do favors to the high-speed circuit operation except that the fully symmetric geometry results in symmetric capacitance, which helps the differential signaling.

For high-speed circuits, the common-centroid guideline is recommended as the global layout guideline, instead of the local optimization. It is verified that all transistors of the delay cell apply a common-centroid scheme, instead of the source coupling scheme, result in a poor performance.

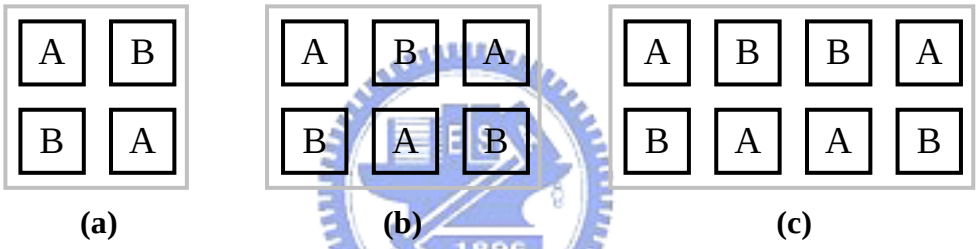



Fig. 6.3: Common-centroid scheme (a) m = 2 (b) m = 3 (c) m = 4

Fig. 6.3 shows the common-centroid scheme for differential signaling. The input transistors are denoted as *A*, and the input-bar transistors are denoted as *B*. Assume the I/O direction is either top-to-bottom or right-to-left. The input capacitance of A equals to that of B, and so does the output capacitance.

## 6.1.2 Grid Design

Fig. 6.4 shows the proposed grid cells for supplies. (a) is the power grid cell. (b) and (c) are the decoupling capacitors filled in the chip's blank area. Cap1 is a stack type capacitor of 0.17 pF. And Cap2 is a finger type capacitor of 0.7 pF.

The grid design is flexible since all the cells behave like tiles. Metals are self-connected, so supplies are easily transported through the cascaded cells.

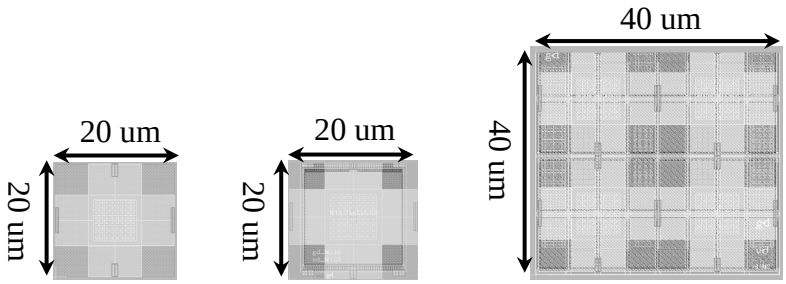



Fig. 6.4: Grid design (a) power grid (b) Cap1 (c) Cap2

### 6.1.3 Chip Layout

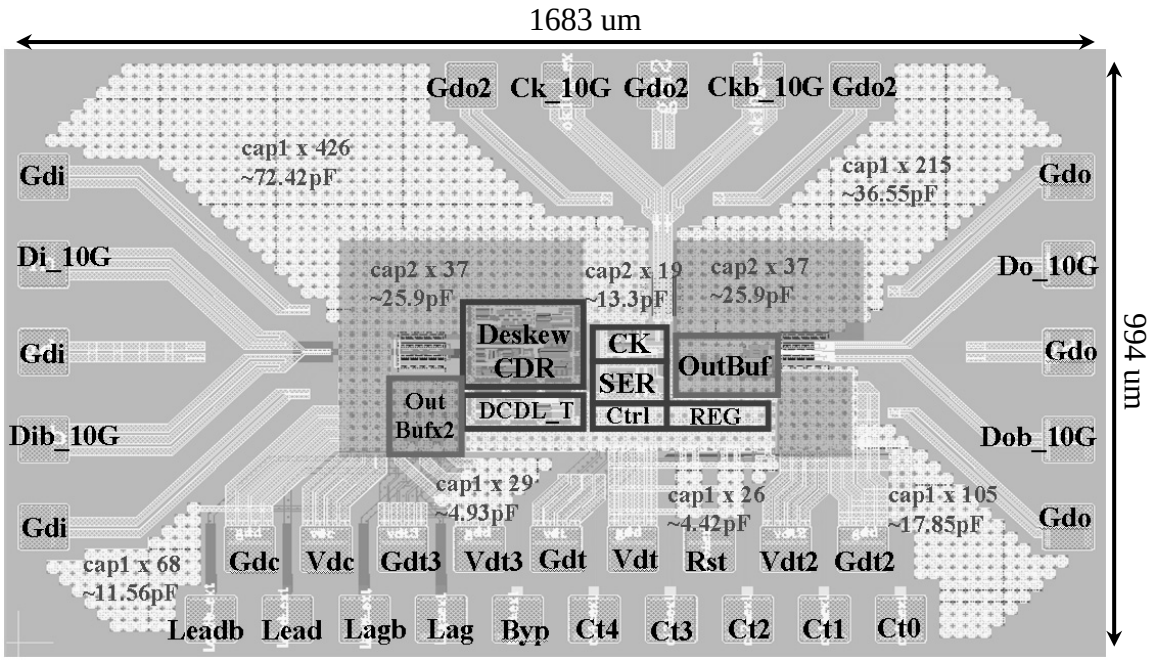



Fig. 6.5: Chip Layout

Table 10: Pads and power configurations

| **Pads Configuration** | |
|---|---|
| G-S-G-S-G Probe | 5 × 3 |
| Vdd / Gnd | 4 × 2 |
| Control | 7 |
| 2.5Gb/s Output | 2 × 2 |

**(a)**

| **Power Configuration** | |
|---|---|
| Vdc, Gdc | 60 mW |
| Vdt, Gdt | 60 mW |
| Vdt2, Gdt2 | 60 mW |
| Vdt2, Gdt2 | 60 mW |

**(b)**

| **Power vs. Modes** | |
|---|---|
| Full | 240 mW |
| Nominal | 180 mW |
| Bypass | 120 mW |
| Debug | 180 mW |

**(c)**

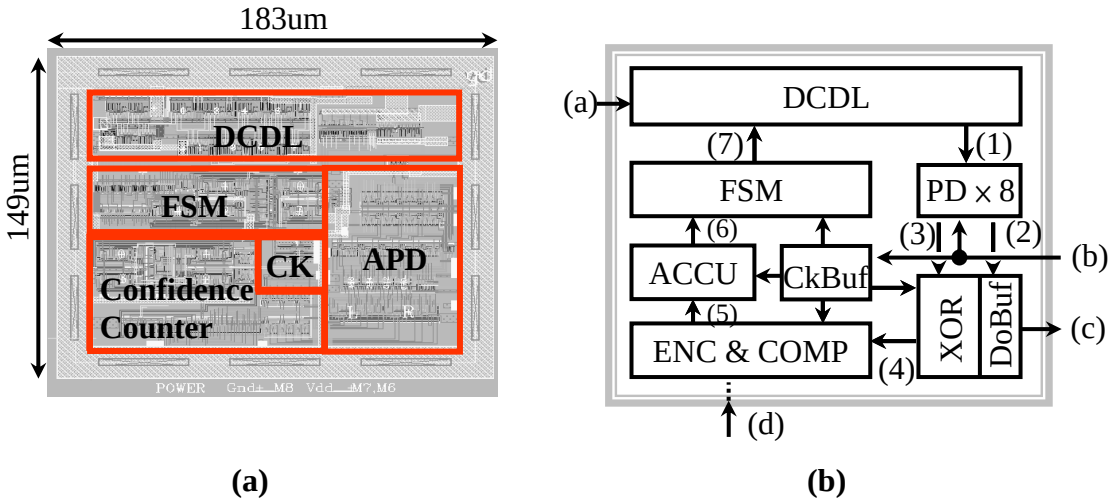
Table 11: Block layout area

| Area (μm × μm) | |
|---|---|
| Full Chip (pads included) | 1683 × 994 |
| Transceiver | 700 × 310 |
| Deskew CDR | 183 × 149 |

The G-S-G-S-G probe pads are used for the high-speed I/O. In Fig. 6.5, there are 3 directions for the 10-G I/O. The data inputs from the left. The clock inputs from the top. The data outputs to the right.

Table 10(b) shows power pads and the corresponding power consumption. Vdt/Gdt power pair is for the serializer, the clock generator, and DCDL_T. Vdt2/Gdt2 is for the 10-Gb/s output buffer. Vdt3/Gdt3 is for the two 2.5-Gb/s output buffers, while Vdc/Gdc is for the CDR. Each power pair provides 60 mW to the internal circuits. The full-chip power consumption depends on the operation modes, shown in Table 10(c). Table 11 shows the layout area. The transceiver area includes all circuits and 3 pairs of 50-ohm termination resistors.

### 6.1.4 Core Layout




**(a)** **(b)**

Fig. 6.6: Deskew CDR (a) layout (b) block layout with I/O ports

Fig. 6.6(a) shows the layout of the CDR. Fig. 6.6(b) shows the block layout and the signal flow. Table 12 shows the ports information. The global reset at port (d) goes to the FSM and the accumulator ACCU. After each burst-data recovery, it resets the

two up-down counters. DCDL is then re-initialized.

Table 12: The ports information of the CDR (a) internal ports (b) external ports

| Internal Ports |
|---|
| (1) DCDL's output Do, Dob |
| (2) Recovered data Q<0,2,4,6> |
| (3) PD's output Q<0:7> |
| (4) XOR's output Lead<0:3>, Lag<0:3> |
| (5) Comparator's output $Lead_{COMP}$, $Lag_{COMP}$ |
| (6) Confidence counter's output $Lead_{CC}$, $Lag_{CC}$ |
| (7) FSM's output C<0:7>, F<0:3> |

**(a)**

| External Ports |
|---|
| (a) 10-Gb/s Input Data |
| (b) Global Clock P<0:7> |
| (c) Recovered Data Do<0:3> |
| (d) Global Reset |

**(b)**

# 6.2 Post Simulation

## 6.2.1 Simulation Setup

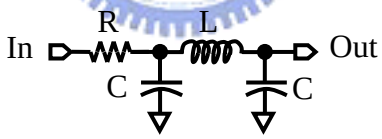



Fig. 6.7: The bonding pad model

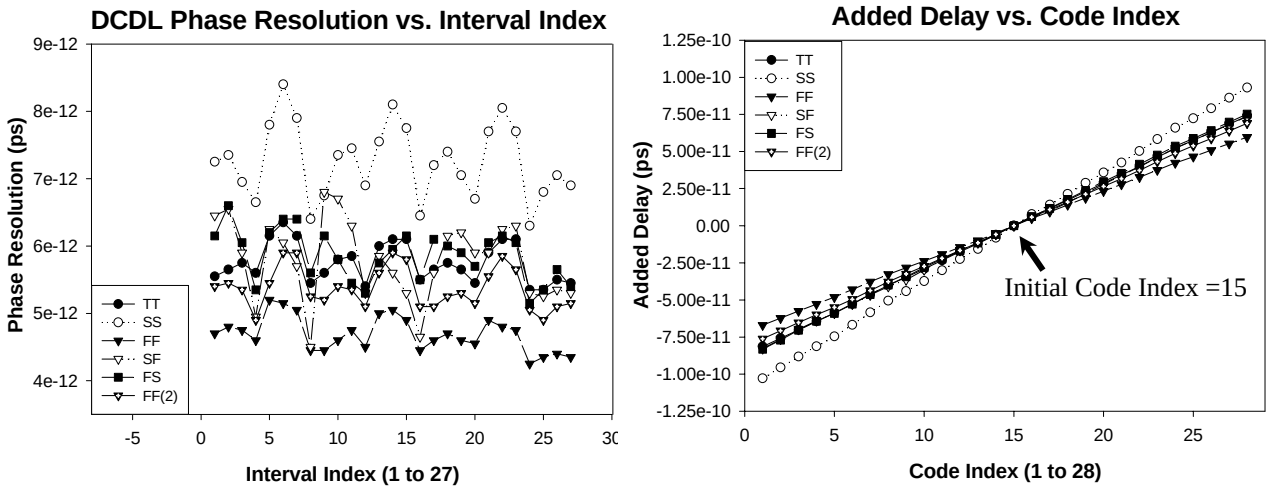
All the mentioned simulations in this section are post simulations. They all adopt the bonding pad model in Fig. 6.7, where R = 1 mΩ, and C = 0.5 pF, and L = 2 nH.

For the simulations of DCDL and the CDR, the parasitic extraction method is R-C-CC. The full-chip simulation uses C-CC instead since the hspice simulator fails to allocate the memory for an extra-large R-C-CC network.

The simulation refers to the typical case by defualt when it is not specified. The typical case refers to the TT-corner model, at room temperature, and under the typical 1.2-V supply.

## 6.2.2 DCDL Simulation

■ **Phase Resolution & Tuning Range**




**(a)** **(b)**

Fig. 6.8: DCDL post simulation (a) phase resolution (b) tuning range

Table 13: DCDL phase resolution and tuning range

| | TT | SS | FF | FF(2) | SF | FS |
|---|---|---|---|---|---|---|
| Avg. Resolution (ps) | 5.76 | 7.25 | 4.69 | 5.37 | 5.79 | 5.86 |
| Tuning Range (ps) | 155.3 | 195.9 | 126.7 | 145.1 | 156.4 | 158.3 |
| Idd (mA, rms) | 18.14 | 13.46 | 24.35 | 16.60 | 18.43 | 17.61 |

* FF(2) simulation is under 1-V supply.

Fig. 6.8(a) shows the post simulation result of DCDL phase resolution. The target average phase resolution is 6 ps. Results of all corner models are given. Fig. 6.8(b) shows the DCDL tuning range. The monotonic tuning behaviors of all corner models can be observed.

The average phase resolution and the tuning range are given in Table 13. In FF-corner case, the '126.7-ps' tuning range is less than the specification '140 ps'. To extend the tuning range as well as to reduce the power, it is a good way to lower down the supply voltage. The FF(2) simulation under 1-V supply proves this concept.

■ **Worst Output Eye**

The worst case of the DCDL output eye occurs at the maximum number of cascaded stages. Fig. 6.9 shows the environment. The longest path is active for the worst eye measurement. The 10-Gb/s input Di is 0.6V±50mV, and the jitter

accumulation time is 1200-bit time, or 120 ns.

Only in the SS-corner case, the peak-to-peak jitter of *14.83 ps* does not meet the specification *10 ps*, or '0.1 UI'. In the FF(2) simulation case, the common mode of the delay cell is self-adjusted, shown in Fig. 6.10(f), because of the CMOS architecture.

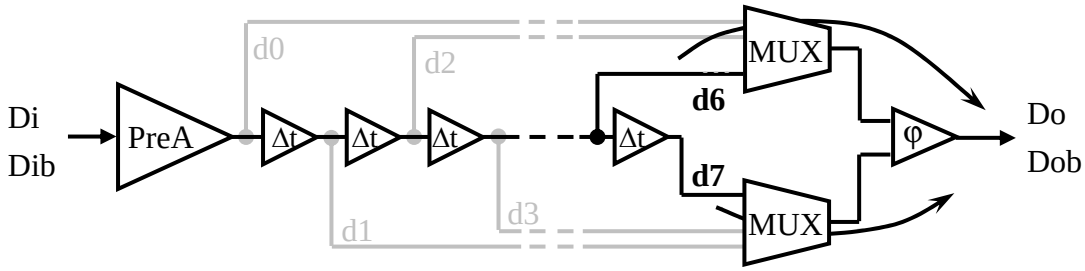



Fig. 6.9: DCDL worst-case setup

Table 14: DCDL output peak-to-peak jitter at the worst-case setup

| | TT | SS | FF | FF(2) | SF | FS |
|---|---|---|---|---|---|---|
| Jp-p (ps) | 4.06 | 14.83 | 4.69 | 3.96 | 3.88 | 8.41 |

* FF(2) simulation is under 1-V supply.

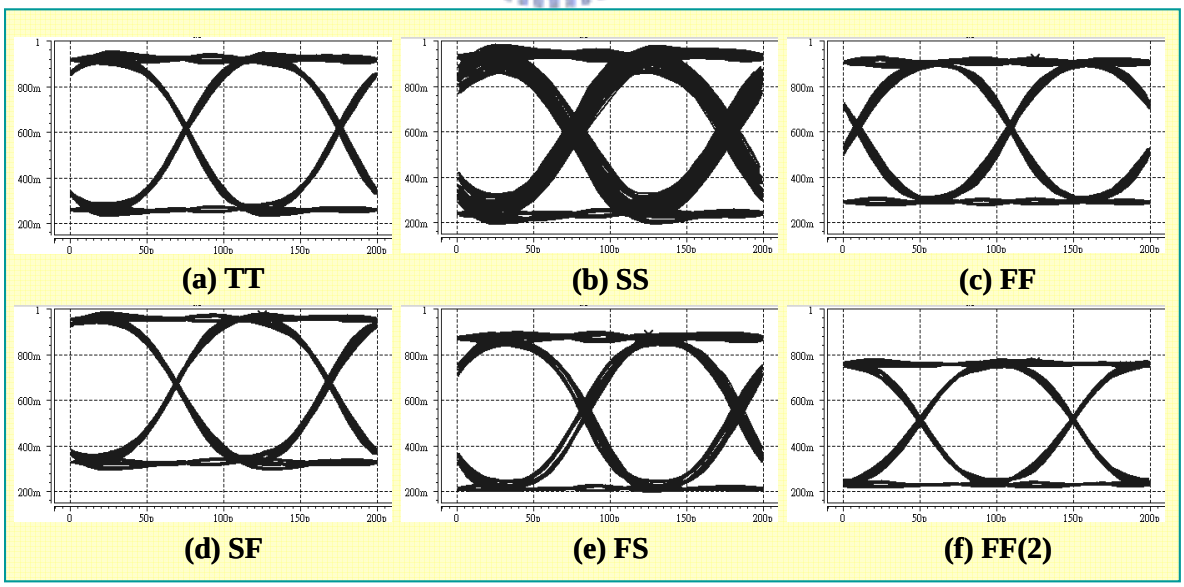


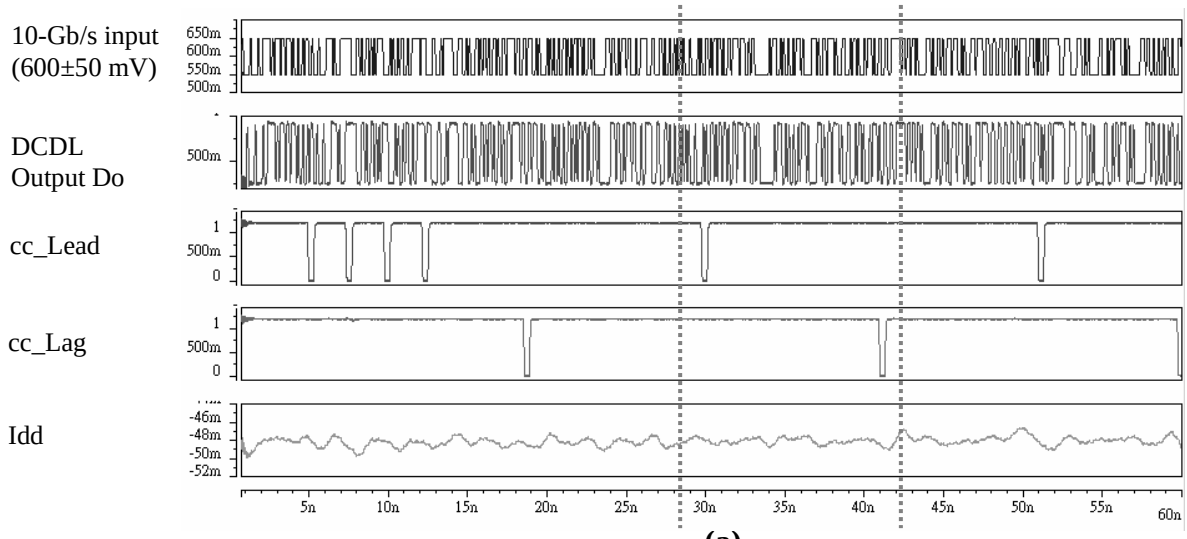

Fig. 6.10: DCDL post simulation - worst eye diagrams

### 6.2.3 CDR Simulation

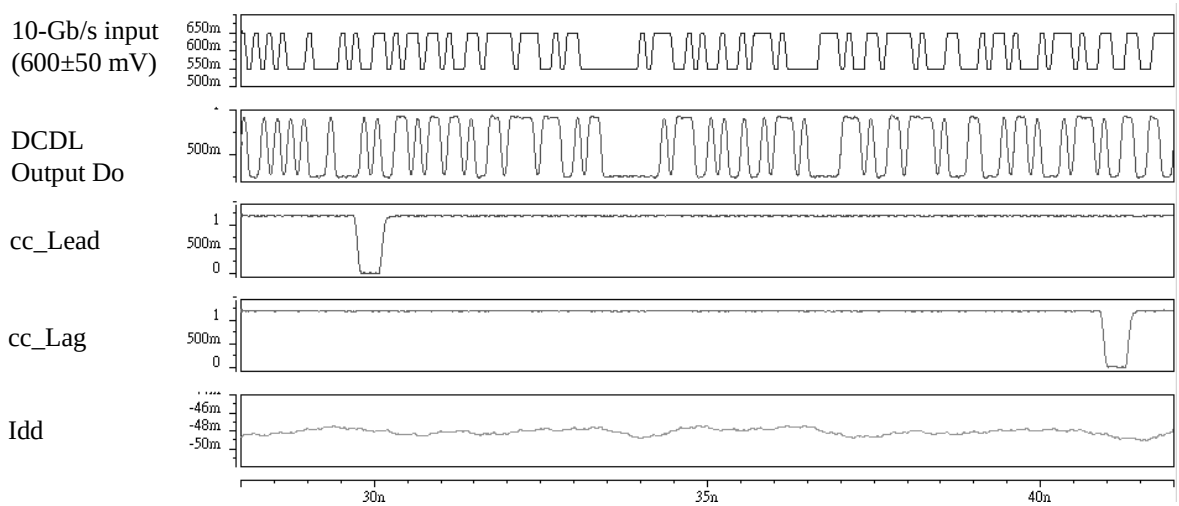
■ **Tracking-to-Locking Simulation**

Fig. 6.11 shows the post simulation of the CDR. In Fig. 6.11(a), it demonstrates the tracking behavior of the '25-ps' average case. The locking behavior is observed after $t = 15ns$.

Fig. 6.11(b) shows a timing interval of Fig. 6.11(a). It observes the relationship between 'the input Di' and 'the output Do' of DCDL. Do is the delayed and amplified version of Di.




**(a)**




**(b)**

Fig. 6.11: Deskew CDR post simulation

■ **FSM Simulation**

Fig. 6.12. follows the simulation in Fig. 6.11(a). The FSM updates the coarse control C<0:7> and fine control F<0:3> according to $Leadb_{CC}$ / $Lagb_{CC}$ of the confidence counter.

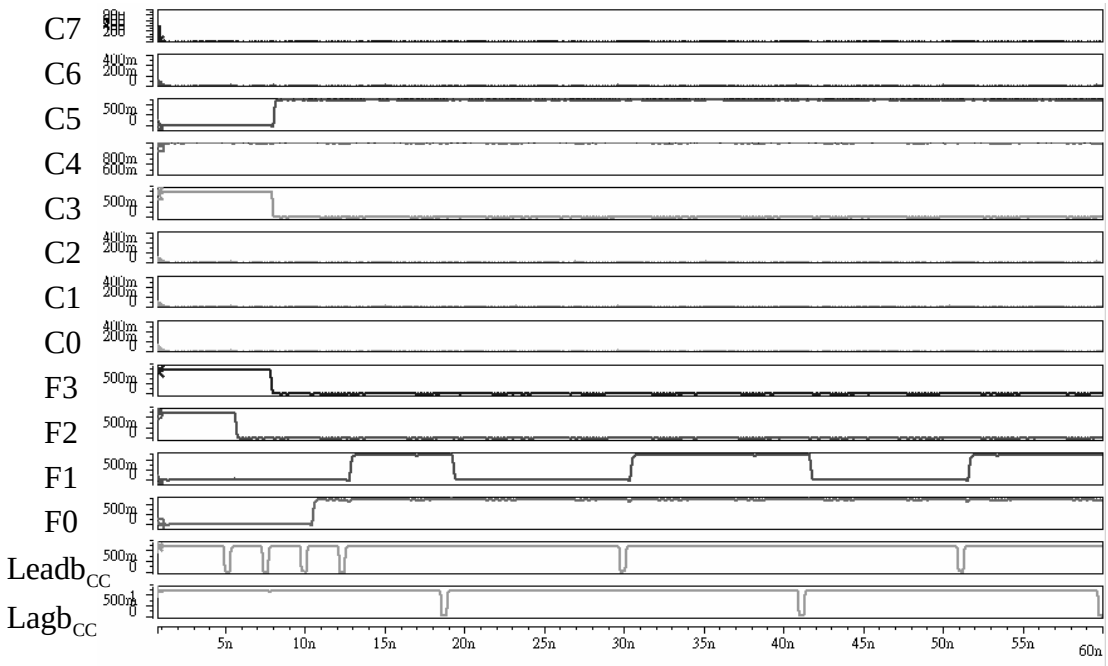
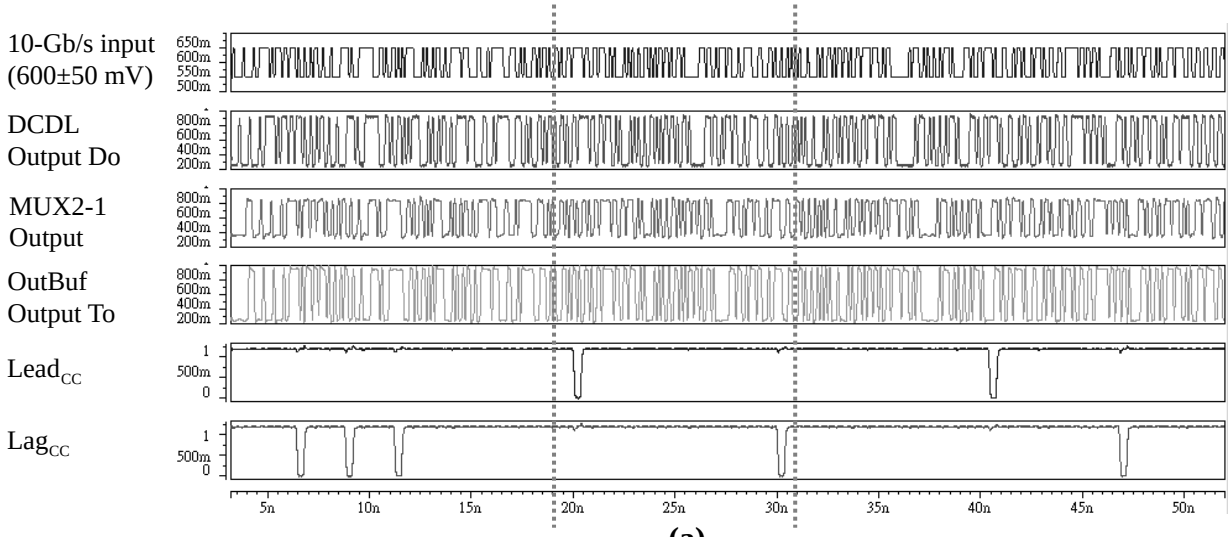



Fig. 6.12: Deskew CDR post simulation – FSM

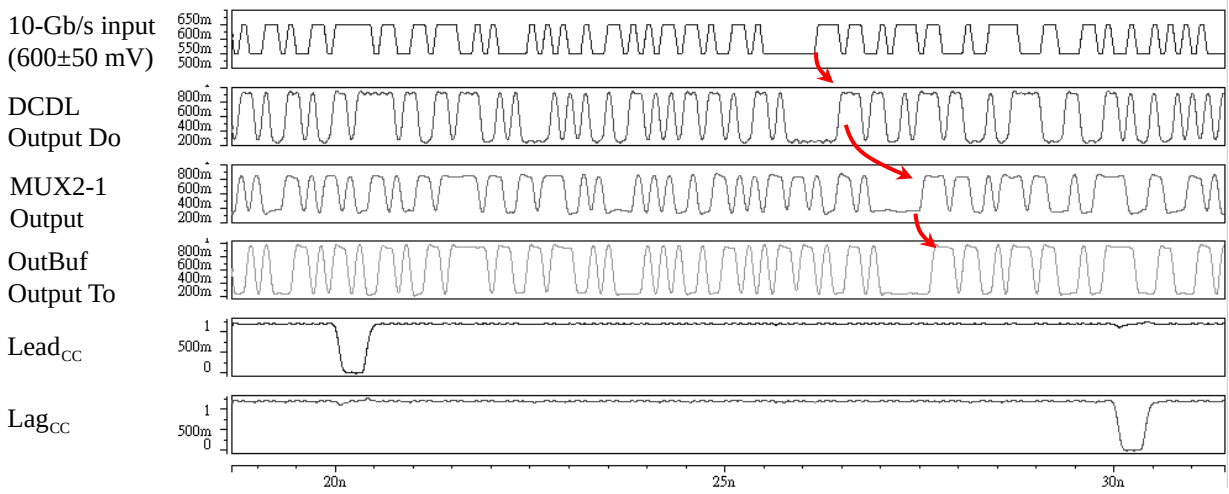
## 6.2.4 Full Chip Simulation

■ **Nominal Operation**

Fig. 6.13 shows the full-chip post simulation in nominal mode. The tracking-to-locking behavior of the CDR is observed. Fig. 6.13(b) is a zoom-in version of (a).

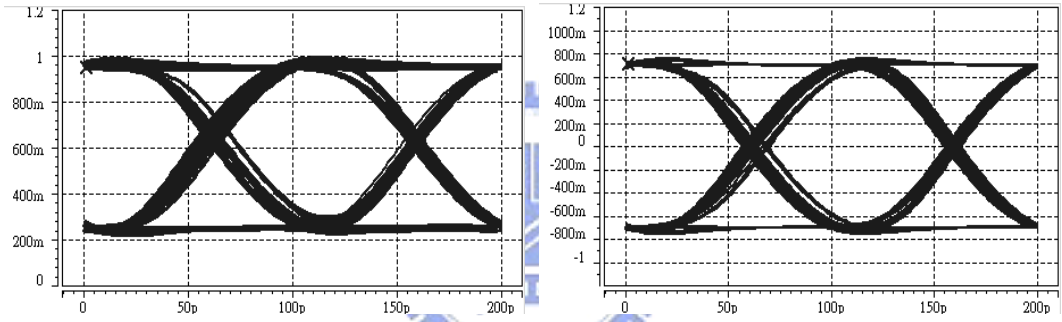



**(a)**




**(b)**

Fig. 6.13: Full-chip post simulation




**(a)** **(b)**

Fig. 6.14: 10-Gb/s output eyes (a) single-ended To (b) differential-ended To-Tob

Fig. 6.14 shows the output eye diagrams of To of 10-Gb/s output buffer. The simulation assumes a capacitive load of 1 pF for each probe pad. The eye diagrams are measured in lock state. Fig. 6.14 (a) shows the single-ended eye of To, and (b) shows the differential-ended eye diagram. Peak-to-peak jitters are *13.4 ps* and *12.8 ps* for case (a) and case (b) respectively.

## 6.3 Test Environment

Agilent N4901B Serial BERT is responsible for the 10-G I/O. It measures the BER and data output eyes, and derives the phase resolution of DCDL. Keithley 2400 source meter provides/measures the power. HP DC Supply provides power to the chip,

of course. It also provides the AC ground, which is half the Vdd level, to the probe pads.

In debug mode, Agilent 86100B oscilloscope monitors the 2.5-Gb/s Lead and Lag of the CDR. It also verifies one of the four recovered output data.

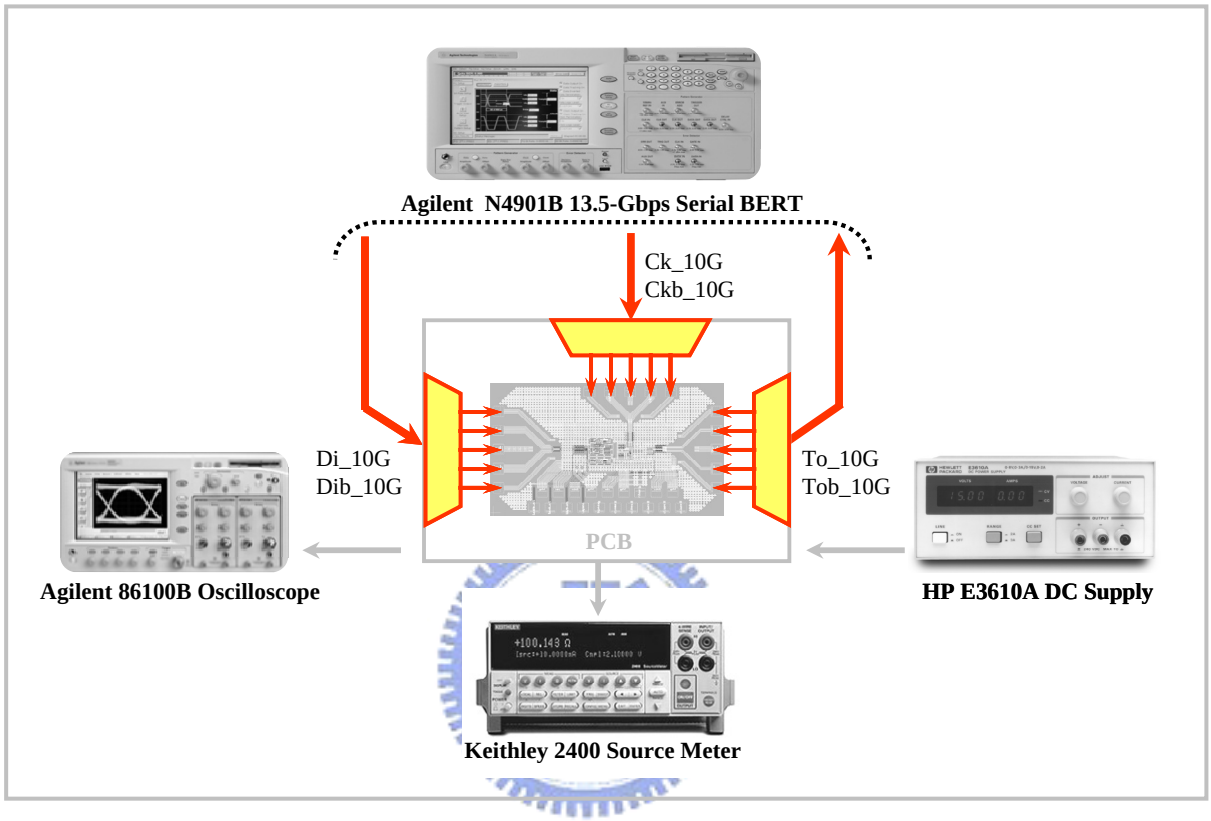



Fig. 6.15: Test environment

# Chapter 7

# Conclusion

In this dissertation, we adopt a data-deskew CDR with advantages of *a simple environment setup*, *low hardware overhead*, and *the easy synchronization for multi-channel timing recovery*. Analysis and design of the CDR are presented in both system and circuit levels. A digital implementation of the 10-Gb/s transceiver is realized in TSMC 0.13μm 1P8M CMOS technology.

In Deskew CDR, the 10-Gb/s input data is modeled as a ±50-mV small signal after the 1-cm channel transmission. The delay cell is not only a deskew buffer but also a pre-amplifier cell. The full-rate data is amplified and delayed in the DCDL. Then it enters the quarter-rate APD. The high-gain APD eliminates the need for a charge pump, while the confidence counter and the FSM serve as a loop filter to the system. Fast acquisition is achieved by the majority-vote scheme in the confidence counter.

Implementation of the CDR adopts digital circuits. In DCDL, the coarse delay is tapped from the delay line, while the fine delay is the interpolating result of the coarse delay. The tuning range of more than 1.4 UI is achieved, where the 1.4 UI is divided

into two parts. The 1-UI delay is for adjusting the sampling point, while the other 0.4-UI delay tracks the phase error, which mainly accumulates from the frequency error, during the 1200-bit recovering process.

In the confidence counter, the counter size N influences the loop bandwidth and the acquisition time directly. N is fixed as 24 in this preliminary design. But it can be programmable as $N \geq 24$. An adaptive loop bandwidth is then introduced. The minimum N is mainly limited by the loop latency in 0.13μm technology, so this implementation adopts the pseudo-NMOS scheme. As the technology advances, digital circuits operate fast enough, and static CMOS logic can be back to the field. The loop latency of the system may no longer dominate the minimum N. Then, a faster acquisition time of the CDR can be achieved.

The serializer and the output buffer also adopt digital circuits. The 10-Gb/s 4-to-1 serializer consists of two-stage multiplexers. The output buffer employs 1-bit delay scheme. The amount of pre-emphasis is determined by the digitally controlled tri-state buffers.

For the consideration of experiments, the phase resolution and the DCDL-induced jitter are measured in bypass mode. The loop-back test of the full-chip transceiver is in nominal mode. Besides, the internal Lead/Lag information and one of the recovered 2.5-Gb/s data can be observed in debug mode.

## 7.1 Specification Table

Table 15 and Table 16 show the specifications of DCDL and the CDR according to the post simulation results.

Table 15: DCDL Specifications

| **DCDL Summary** | |
|---|---|
| Input Sensitivity | 100mVp-p |
| Power Consumption | 21.83 mW |
| Average Phase Resolution | 5.76 ps |
| Tuning Range | 1.55 UI |
| Worst Peak-to-Peak Jitter | < 5 ps |

Table 16: Deskew CDR Specifications

| **Deskew CDR Summary** | |
|---|---|
| Technology | TSMC 0.13-μm CMOS 1P8M |
| Input Data Rate | 10 Gb/s |
| Supply Voltage | 1.2 V |
| Power Consumption | 58.2 mW typ. |
| Core Area | 149 μm x 183 μm |
| Acquisition Time | |
| (1) Average Acquisition Time | 11.11 ns |
| (2) Worst Acquisition Time | 22.22 ns |
| Allowed Jp-p Between TX & RX | 0.55 UI |
| Counter Size N | 24 |
| Frequency Tolerance | 1807.9 ppm |
| Loop Bandwidth | 14.39 Mhz |

## 7.2 Comparison

Fig. 7.1 compares the power consumption among several 10-Gb/s CDR systems in different technologies, where the naming notation follows Table 17.

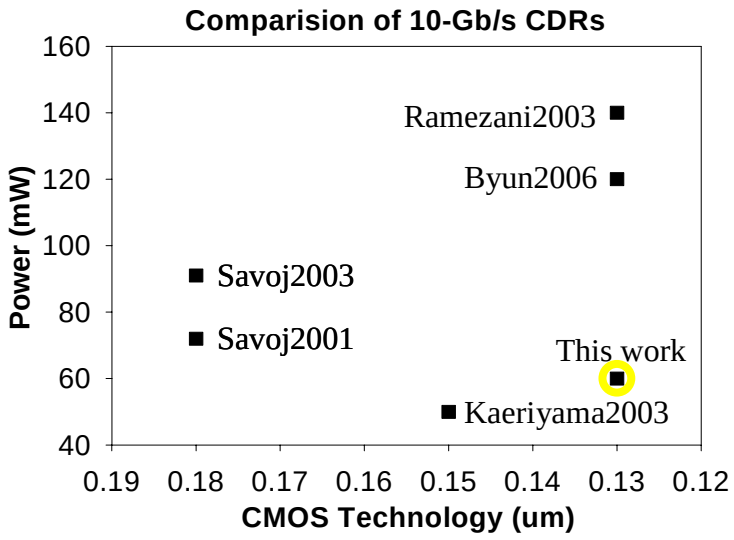



Fig. 7.1: Power comparison for 10-Gb/s CDRs

Deskew CDR uses active loads for the inductive peaking scheme. Although the power is a trade-off item for the bandwidth, it still operates in the competitive power consumption of 60 mW.

Table 17: A comparison of multi-gigabit CDR systems

| CDR Systems | **D**ata Rate | CMOS Tech. | **P**ower (mW) | **A**rea (mm^2) | Pub. | **D/P** | **D/P/A** |
|---|---|---|---|---|---|---|---|
| Ohtomo2006 [9] | 12.5Gb/s | 013 $\mu$ m | 400 | 1.5 × 0.75 | ISSC JNL | 0.03 | 0.03 |
| Byun2006 [8] | 10Gb/s | 013 $\mu$ m 1P8M | 120 (Core) | N/A | ISSC JNL | 0.08 | - |
| Kreienkamp2005 [16] | 10.8Gb/s | 011 $\mu$ m | 220 | 0.25 × 1.4 | ISSC JNL | 0.05 | 0.14 |
| Farjad-Rad2004 [30] | 8Gb/s | 013 $\mu$ m | 33 | 0.28 × 0.28 | ISSC JNL | 0.24 | 3.09 |
| Savoj2003 [6] | 10Gb/s | 018 $\mu$ m | 91 | 1.75 × 1.55 | ISSC JNL | 0.11 | 0.04 |
| Kaeriyama2003 [21] | 10Gb/s | 015 $\mu$ m | 50 | 0.12 × 0.13 | ISSC CNF | 0.2 | 12.82 |
| Ramezani2003 [31] | 10Gb/s | 013 $\mu$ m | 140 | 1.2 × 1.2 | ISCAS CNF | 0.07 | 0.05 |
| Savoj2001 [4] | 10Gb/s | 018 $\mu$ m | 72 | 1.1 × 0.9 | ISSC JNL | 0.14 | 0.14 |
| This Work | 10Gb/s | 013 $\mu$ m 1P8M | 60 | 0.18 × 0.15 | | 0.17 | 6.17 |

# Bibliography

[1] Özgür Çobanoğlu, "CDR Status Report", INFN & Univ. of Turin, Turin, Italy. Nov 2006. http://www.ph.unito.it/~cobanogl/CDR/cdr_cern_mic-dg.pdf

[2] Behzad Razavi, *Design of Integrated Circuits for Optical Communications*, pp.322-327, McGRAW- Hill, 2003.

[3] Charles R. Hogge, JR., "A Self Correcting Clock Recovery Circuit", IEEE JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. LT-3, NO. 6, Dec 1985.

[4] Jafar Savoj and Behzad Razavi, "A 10-Gb/s CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 36, NO. 5, May 2001.

[5] Jri Lee and Behzad Razzavi, "A 40-Gb/s Clock and Data Recovery Circuit in 0.18-μm CMOS Technology", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 12, Dec 2003.

[6] Jafar Savoj and Behzad Razavi, "A 10-Gb/s CMOS Clock and Data Recovery Circuit With a Half-Rate Binary Phase/Frequency Detector", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 1, Jan 2003.

[7] Lindor Henrickson, David Shen, Uno Nellore, Alan Ellis, Joong Oh, HuiWang, Giovanni Capriglione, Ali Atesoglu, Alice Yang, Peter Wu, Syed Quadri, and David Crosbie, "Low-Power Fully Integrated 10-Gb/s SONET/SDH Transceiver in 0.13-μm CMOS", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 10, Oct 2003.

[8] Sangjin Byun, Jyung Chan Lee, Jae Hoon Shim, Kwangjoon Kim, Member, IEEE, and Hyun-Kyu Yu, "A 10-Gb/s CMOS CDR and DEMUX IC With a Quarter-Rate Linear Phase Detector", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 11, Nov 2006.

[9] Yusuke Ohtomo, Kazuyoshi Nishimura, and Masafumi Nogawa, "A 12.5-Gb/s Parallel Phase Detection Clock and Data Recovery Circuit in 0.13-μm CMOS", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 9, Sep 2006.

[10] Sungjoon Kim, Kyeongho Lee, Deog-Kyoon Jeong,David D. Lee, and Andreas G. Nowatzyk, "An 800Mbps Multi-Channel CMOS Serial Link with 3x Oversampling", IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE 1995.

[11] Chih-Kong Ken Yang, “DESIGN OF HIGH-SPEED SERIAL LINKS IN CMOS”, Ph. D. dissertation, pp. 120-128, Stanford, Dec 1998.

[12] S. I. Ahmed, Tad A. Kwasniewski, "OVERVIEW OF OVERSAMPLING CLOCK AND DATA RECOVERY CIRCUITS", CCECE/CCGEI, Saskatoon, May 2005.

[13] Jaeha Kim and Deog-Kyoon Jeong, "Multi-Gigabit-Rate Clock and Data Recovery Based on Blind Oversampling", IEEE Communications Magazine, Dec 2003.

[14] Chih-Kong Ken Yang, Ramin Farjad-Rad, and Mark A. Horowitz, "A 0.5- m CMOS 4.0-Gbit/s Serial Link Transceiver with Data Recovery Using Oversampling", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 33, NO. 5, May 1998.

[15] M.-J. Edward Lee, William J. Dally, John W. Poulton, Patrick Chian, and Stephen F. Greenwood', "An 84-mW 4-Gb/s Clock and Data Recovery Circuit for Serial Link Applications", 2001 Symposium on VLSl Circuits Digest of Technical Papers.

[16] Rainer Kreienkamp, Ulrich Langmann,Christoph Zimmermann, Takuma Aoyama, and Hubert Siedhoff, "A 10-Gb/s CMOS Clock and Data Recovery Circuit With an Analog Phase Interpolator", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 40, NO. 3, Mar 2005.

[17] Thomas Y. K. Wong and John E. Sitch, "A 10 Gb/s ATM Data Synchronizer", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 31, NO. IO, Oct 1996.

[18] Hung-Wen Lu, Yin-Tin Chang, Chau-Chin Su, "All Digital 625Mbps & 2.5Gbps Deskew Buffer Design", IEEE CNF VDAT Apr 2005.

[19] Makoto Nakamura, Noboru Ishihara, and Yukio Akazawa, "A 156 Mbps CMOS Clock Recovery Circuit for Burst-mode Transmission", 1996 Symposium on VLSl Circuits Digest of Technical Papers.

[20] Masafumi Nogawa, Kazuyoshi Nishimura, Shunji Kimura,Tomoaki Yoshida, Tomoaki Kawamura, Minoru Togashi,Kiyomi Kumozaki, Yusuke Ohtomo, "A 10Gb/s Burst-Mode CDR IC in 0.13µm CMOS", 2005 IEEE International Solid-State Circuits Conference, Session 12.5.

[21] Shunichi Kaeriyama, Masayuki Mizuno, "A 10Gb/s/ch 50mW 120x130µm$^2$ Clock and Data Recovery Circuit", 2003 IEEE International Solid-State Circuits Conference, Session 4.1.

[22] Behnam Analui and Ali Hajimiri, "Instantaneous Clockless Data Recovery and Demultiplexing", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 52, NO. 8, Aug 2005.

[23] Marcus van Ierssel, Ali Sheikholeslami, Hirotaka Tamura,William W. Walker,"A 3.2Gb/s Semi-Blind-Oversampling CDR", IEEE International Solid-State Circuits Conference 2006.

[24] Thomas H. Lee, and John F. Bulzacchelli, "A 155-MHz Clock Recovery Delay- and Phase-Locked Loop", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 27, NO. 12, Dec 1992.

[25] Yong-Cheol Bae and Gu-Yeon Wei, "A MIXED PLL/DLL ARCHITECTURE FOR LOWJITTER CLOCK GENERATION", ISCAS 2004.

[26] G.-Y. Wei, J. Stonick, D. Weinlader, J. Sonntag, and S. Searles, “A 500-MHz MP/DLL clock generator for a 5Gb/s backplane transceiver in 0.25μm CMOS,” IEEE Solid-State Circuits Conference Digest of Technical Papers, pp. 464–465, Feb 2003.

[27] Hiok-Tiaq Ng, Ramin Farjad-Rad, M.-J. Edward Lee, William J. Dally, Trey Greer, John Poulton, John H. Edmondson, Rohit Rathi, and Ramesh Senthinathan, “A Second-Order Semidigital Clock Recovery Circuit Based on Injection Locking”, IEEE JNL Solid-State Circuits, Vol. 38, No. 12, Dec 2003.

[28] Ming-Ju Edward Lee, “AN EFFICIENT I/O AND CLOCK RECOVERY DESIGN FOR TERABIT INTEGRATED CIRCUITS”, Ph. D. dissertation, Stanford, pp. 86-87, Aug 2001.

[29] J. D. H. Alexander, "Clock Recovery from Random Binary Data", Electronics Letters vol. 11, pp. 541-542, Oct 1975.

[30] Ramin Farjad-Rad, Anhtuyet Nguyen, James M. Tran, Trey Greer, John Poulton, William J. Dally, John H. Edmondson, Ramesh Senthinathan, Rohit Rathi, M.-J. Edward Lee, and Hiok-Tiaq Ng, "33-mW 8-Gb/s CMOS Clock Multiplier and CDR for Highly Integrated I/Os", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 39, NO. 9, Sep 2004.

[31] Mehrdad Ramezani and C. Andre. T. Salama, "A l0Gb/s CDR with a Half-rate Bang-Bang Phase Detector", IEEE CNF ISCAS May 2003.