Ph.D.Dissertation

Design of Maximum-Eye-Tracking CDR with Biased Data-Level and Eye Slope Detector for Near-Optimal Timing Adaptation

최적에 가까운 타이밍 적응을 위해
치우친 데이터 레벨과 눈 경사 디텍터를 사용한
최대 눈크기추적 클럭 및 데이터 복원회로 설계

by

Hye-Yoon Joo

February, 2021

Department of Electrical and Computer Engineering
College of Engineering
Seoul National University

Design of Maximum-Eye-Tracking CDR with Biased Data-Level and Eye Slope Detector for Near-Optimal Timing Adaptation

지도 교수 정 덕 균

이 논문을 공학박사 학위논문으로 제출함
2021 년 2 월

서울대학교 대학원
전기·정보공학부
주 혜 윤

주혜윤의 박사 학위논문을 인준함
2021 년 2 월

위 원 장 김 재 하 (인)

부위원장 [illegible] (인)

위 원 이 강 윤 (인)

위 원 전 정 훈

위 원 최 우 기

Design of Maximum-Eye-Tracking CDR with Biased Data-Level and Eye Slope Detector for Near-Optimal Timing Adaptation

by
Hye-Yoon Joo

A Dissertation Submitted to the Department of
Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
at
SEOUL NATIONAL UNIVERSITY
February, 2021

Committee in Charge:

Professor Jaeha Kim, Chairman

Professor Deog-Kyoon Jeong, Vice-Chairman

Professor Kang-Yoon Lee

Professor Jung-Hoon Chun

Professor Woo-Seok Choi

Abstract

In this thesis, design of a maximum-eye-tracking CDR (MET-CDR) for minimum bit error rate (BER) is proposed. The proposed CDR does not require a BER counter or an eye-opening monitor with any iterative procedure to find the near-optimal sampling phase. The biased data-level obtained from the weighted sum of error sampler outputs, UP and DN, extracts the actual eye height information in the presence of pre-cursor ISI. Two samplers operating on two slightly different timings detect the current eye height and the polarity of the eye slope so that the CDR tracks the maximum eye height where the slope becomes zero. Measured results show that the sampling phase of the maximum eye height and that of the minimum BER match well. A prototype receiver fabricated in 28 nm CMOS process operates at 26 Gb/s with an eye-opening of 0.25 UI and consumes 87 mW while equalizing 23.5 dB of loss at 13 GHz.

Keywords: Bit error rate (BER), clock and data recovery (CDR), decision feedback equalizer (DFE), high-speed links, pre-cursor intersymbol interference (ISI), sampling point control, SS-LMS algorithm, timing adaptation.

Student Number: 2016-30218

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Clock and data recovery (CDR) circuits are essential in many high-speed serial link applications. Traditional CDR techniques such as bang-bang CDR (BB-CDR) [1] are widely used because of the simplicity of hardware implementation [2] , [3] . With data and edge samples, the BB-CDR converges to a point where the average value at the edge sampling phase becomes zero. That is, when the main cursor is h_0 and two edge cursors are defined as $h_{+0.5}$ and $h_{-0.5}$, the BB-CDR locks at the phase where $h_{+0.5}=h_{-0.5}$. Fig. 1 shows the single bit response (SBR) of a channel and the locked phase with the BB-CDR. Due to the asymmetric shape of a typical SBR, the data sampling phase or the locked phase with the BB-CDR usually appears behind the peak of the SBR. In other words, the loop does not converge to the minimum bit

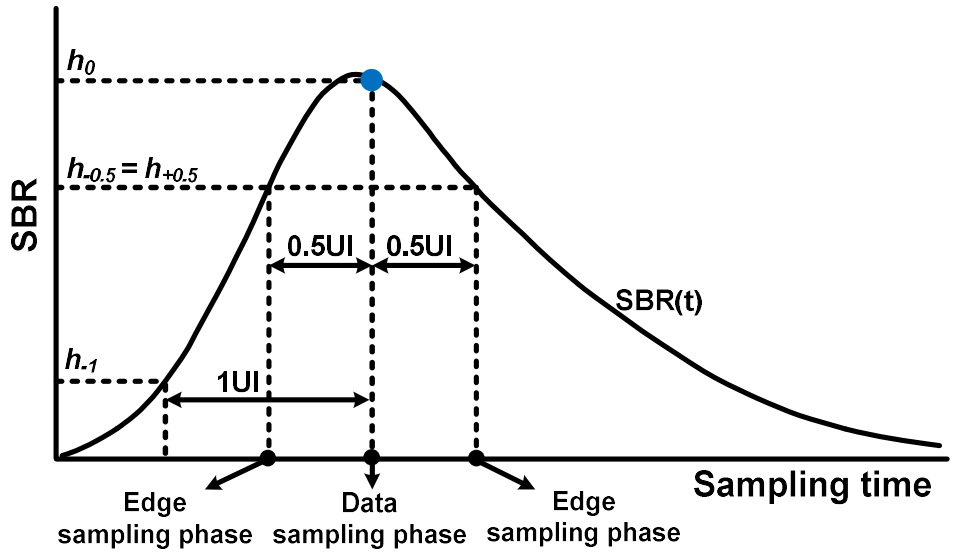


Fig. 1.1 SBR of a channel and its locked phase with BB-CDR.

error rate (BER) sampling phase. Moreover, the eye diagram could even be closed with large pre-cursor intersymbol interference (ISI).

There are several approaches for optimal clock recovery. Shifting the sampling phase from the nominal position effectively improves the BER by reducing the influence of the pre-cursor ISI [4] , [5] . However, the amount of phase shift is determined by using BER estimation that requires either a large silicon area with a long

test time or an off-chip assistance from an external controller. Although stochastic hill-climbing algorithm adopted in [5] reaches the optimum with fewer iterations than the basic hill-climbing algorithm in [4] , measuring the BER for each stage is time-consuming, especially for low BER goals. Other works based on the eye-opening monitor (EOM) [6] , [7] avoid obtaining accurate BER by defining indirect measures: the code mismatch error rate (CMER) with a specified reference voltage [6] or a predefined error count based on the standard deviation of the probability distribution function (PDF) [7] . While the indirect criteria require a shorter processing time than the BER counting in finding the optimal phase, EOM-based CDRs still require complex hardware and long processing time because they operate iteratively by sweeping the sampling phase.

To further simplify the optimal clock recovery, we propose a maximum-eye-tracking CDR (MET-CDR) [8] . The sampling phase for the maximum vertical eye margin, the maximum horizontal eye margin, and the minimum BER may not perfectly coincide, varying over the channel characteristics. In this work, the vertical eye margin is used as a criterion for near-optimal sampling phase. The basic concept is based on the analysis that the settings for the maximum vertical eye margin and the minimum BER match very well [4] . By tracking the maximum eye height, the proposed CDR effectively finds the optimal sampling phase with simple hardware and short processing time, not requiring BER counting, EOM, nor any iterative procedure.

1.2 Thesis Organization

This thesis is organized as follows. In Chapter2, backgrounds of the design of the receiver for high speed links are explained. The basic operation and building blocks such as equalizer and CDR of the general receiver front-end are provided. And, the previous architectures of the CDR searching optimal sampling phase are introduced to show the motivation of this work. The comparison and limitation of the previously proposed CDRs are presented.

In Chapter3, a maximum-eye-tracking CDR is presented. The concept of biased data-level for eye height information, the eye slope detector and the adaptation algorithm are explained. The whole architecture and implementation are shown. Then, the verification of the algorithm with simulations results are shown. Further analysis on the biased data-level such as algorithm accuracy and effect of variables are also given in this chapter. At the end of this chapter, the expansion of the proposed CDR to PAM4 signaling is described.

In Chapter4, the measurement results are presented. The data-levels as changing the variables and the bathtub curve results are measured to estimate the optimal sampling phase. Also, the jitter tolerance curves are measured as varying the test options.

Chapter5 summarizes the proposed works and concludes this thesis.

Chapter 2

Backgrounds

2.1 Receiver Front-End

Fig. 2.1 shows a simplified representation of the general transceiver's I/O interface. Specifications may vary depending on various applications, the ultimate purpose of all transceiver interfaces is to send and receive data so that it has an error rate lower than the target BER [9] -[11] . To achieve the desired performance, there are several roles for receiver front-end. The characteristics of channels should be considered and the loss of channel should be compensated through equalizers [12] -[14] . Clock with low phase noise is required and the sampling position also has a significant impact [15] -[17] .

In this chapter, the characteristic of the channel and system estimation from channel are given. Operation and analysis of typical equalization schemes and CDR are also explained.

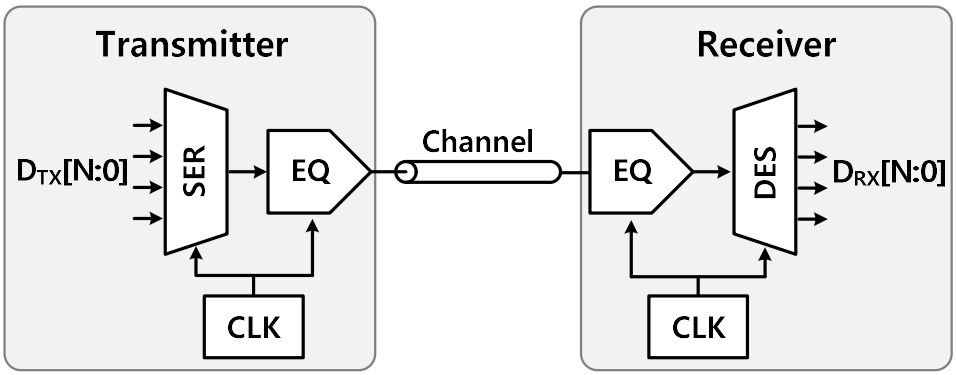
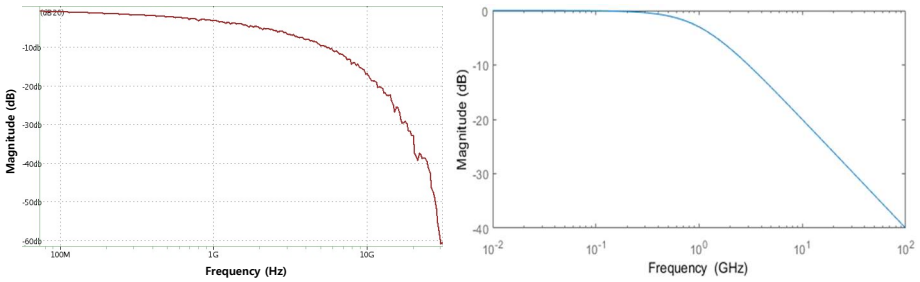


Fig. 2.1 I/O interface of typical transceiver.

2.1.1 Channel

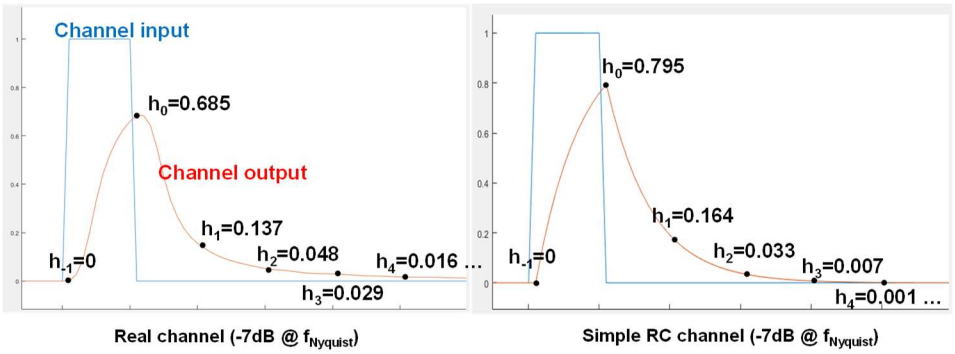
2.1.1.1 Channel Characteristics

Fig. 2.2(a) and (b) show the S21 of a real channel and simple RC channel, respectively. In real channel, S21 is affected by skin effect of conductor, dielectric loss of insulator, reflection, and so on [18] . On the other hand, in the RC channel with one pole, the magnitude is simply decaying with a slope of -20dB/dec. Although the simple RC channel excludes the phenomena that occur in the real environment, but is good for intuitive analysis. Fig. 2.3 shows the single bit responses and resulting eye diagrams with PRBS pattern. In these figure, it is assumed that h_0 is determined at the peak of SBR.

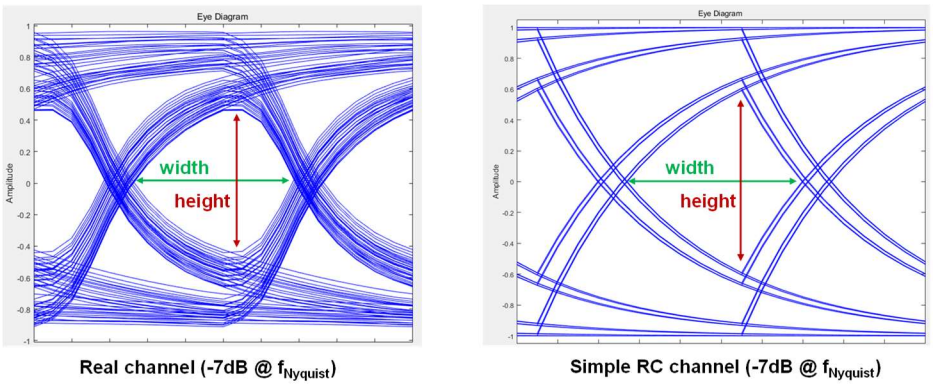


(a) (b)

Fig. 2.2 S21 of (a) real channel and (b) simple RC channel.



(a)



(b)

Fig. 2.3 (a) Single bit response and (b) eye diagram with real channel and simple RC channel.

2.1.1.2 Maximum Data-Rate Estimation with Channel Characteristics

Analysis from now on is based on RC channel with pole at 1GHz. We can calculate vertical and horizontal eye opening for various data-rates. By looking at the channel loss at the target frequency, much of the system design can be predicted.

For vertical eye height, the worst case pattern – isolated +1 or -1 – should be considered as shown in Fig. 2.4(a). With the step response, vertical eye opening can be represented as follows:

$$\boldsymbol{Eye_{vertical} = V_o\left(1 - e^{-\frac{t_o}{RC}}\right) - V_o e^{-\frac{t_o}{RC}} = V_o\left(1 - 2e^{-\frac{t_o}{RC}}\right)} \tag{2.1}$$

where

$$\boldsymbol{t_o = 1UI.} \tag{2.2}$$

The relationship between vertical eye opening and Nyquist frequency is plotted in Fig. 2.4(b). Assuming vertical eye opening of 90%, the channel loss is about -3.2dB, and it means that Nyquist frequency is similar to corner frequency of the channel.

For horizontal eye width as shown in Fig. 2.5(a), we can use two expressions at t_1 and t_2 as follows:

$$\boldsymbol{0.5V_o = V_o\left(1 - e^{-\frac{t_1}{RC}}\right)} \tag{2.3}$$

and

$$\boldsymbol{0.5V_o = V_o\left(1 - e^{-\frac{t_2}{RC}}\right) - V_o\left(1 - e^{-\frac{t_2 - t_o}{RC}}\right)} \ . \tag{2.4}$$

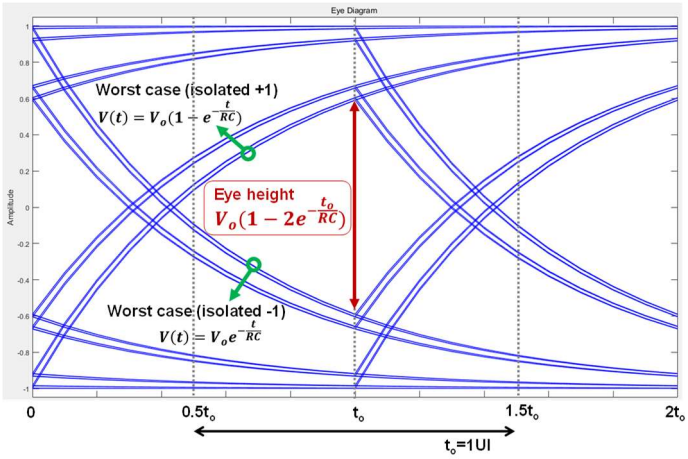
From (2.4), we can calculate t_2 as follows:

$$\boldsymbol{t_2 = \left[ln\left(e^{\frac{t_o}{RC}} - 1 \right) - ln0.5 \right] RC} \quad (2.5)$$

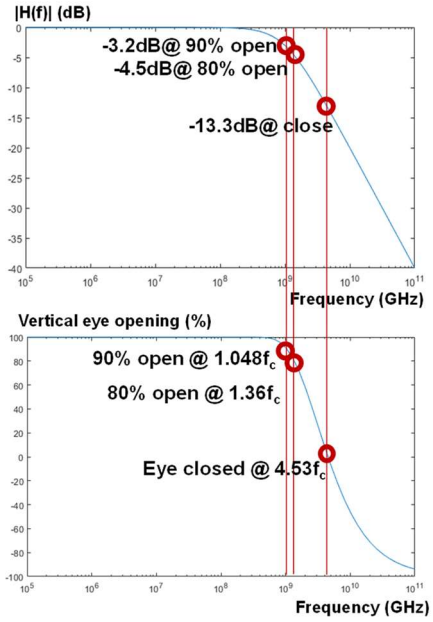
and the resulting horizontal eye width is determined as follows:

$$\boldsymbol{t_2 - t_1 = \left[ln\left(e^{\frac{t_o}{RC}} - 1 \right) \right] RC}. \quad (2.6)$$

Fig. 2.5(b) shows the relationship between horizontal eye opening and Nyquist frequency. For eye width of 90%, the Nyquist frequency should be less than 2 times of corner frequency. Fig. 2.5(c) shows the comparison between vertical and horizontal eye opening (%) according to frequency. For the same data rate, vertical eye opening decreases faster. So it is important to consider eye height when estimating the maximum data rate that can be operated for a given channel.

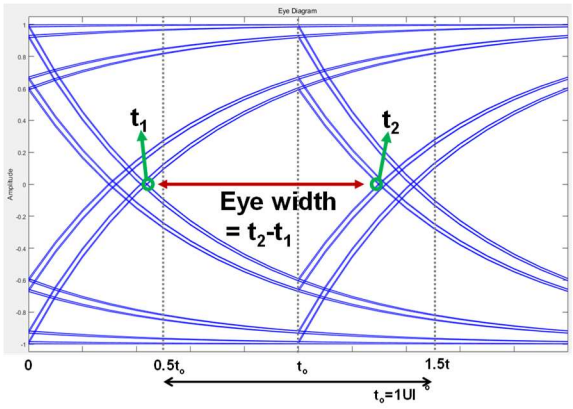


(a)

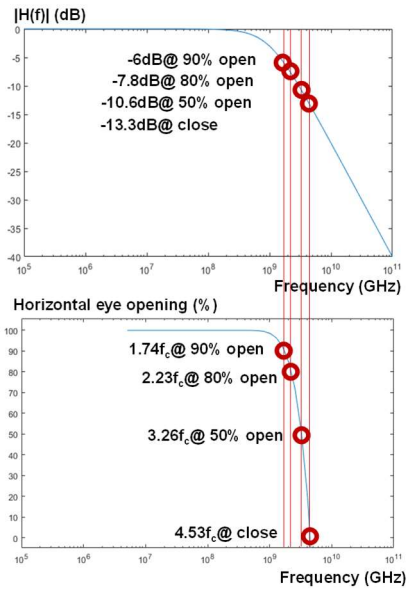


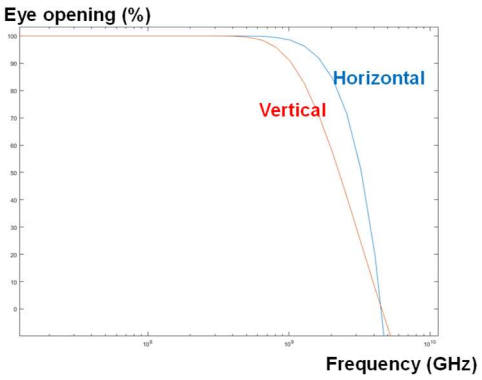
(b)

Fig. 2.4 (a) Vertical eye opening with worst case. (b) Intuitive relation between symbol rate vs. vertical eye opening in RC channel.



(a)





(b) (c)

Fig. 2.5 (a) Horizontal eye opening with worst case. (b) Intuitive relation between symbol-rate vs. horizontal eye opening in RC channel. (c) Comparison between vertical and horizontal eye opening.

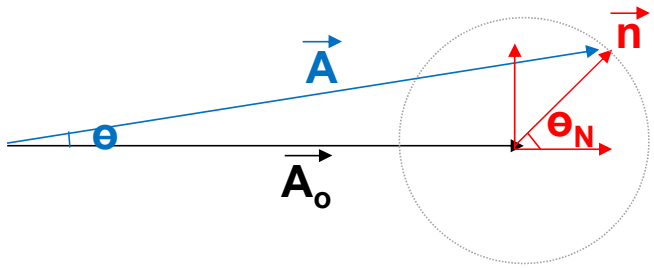


Fig. 2.6 Signal and noise in phasor domain.

For a more accurate analysis, we can add the influence of additive white Gaussian noise (AWGN). Fig. 2.6 shows a phasor domain expression of signal and noise [19] . Noise can be expressed as follows:

$$n\angle\theta_N = n\cos\theta_N + jn\sin\theta_N \quad (2.7)$$

and the rms power is as follows:

$$\sigma_N^2 = \frac{n^2\cos^2\theta_N}{2} + \frac{n^2\sin^2\theta_N}{2} \ . \quad (2.8)$$

Signal affected by noise and the magnitude of the signal can be written as

$$A\angle\theta = A_o\angle 0 + n\angle\theta_N = (A_o + n\cos\theta_N) + jn\sin\theta_N \ , \quad (2.9)$$

$$|A|^2 = (A_o + n\cos\theta_N)^2 + (n\sin\theta_N)^2$$

$$\cong (A_o + n\cos\theta_N)^2 \cong A_o^2 + n^2\cos^2\theta_N \ , \quad (2.10)$$

and

$$\boldsymbol{amp_{rms}^2 = \frac{A_o^2}{2} + \frac{n^2 cos^2 \theta_N}{2}} \quad . \qquad (2.11)$$

The degree of signal can be written as

$$\boldsymbol{\theta = arctan\left(\frac{nsin\theta_N}{A_o + ncos\theta_N}\right) \cong \frac{nsin\theta_N}{A_o + ncos\theta_N} \cong \frac{nsin\theta_N}{A_o}} \quad , \qquad (2.12)$$

so the resulting rms power in jitter is

$$\boldsymbol{jitter_{rms}^2 = \left(\frac{\theta_{rms}}{2\pi T}\right)^2 = \left(\frac{1}{2\pi f}\right)^2 \theta_{rms}^2 = \left(\frac{1}{2\pi f}\right)^2 \frac{n^2 sin^2 \theta_N}{2A_o^2}}. \qquad (2.13)$$

From (2.8), (2.11) and (2.13), we can conclude that the noise power affects both amplitude and jitter. The part of the random noise that matches the current signal phase is converted to amplitude noise. The quadrature component of the noise does not affect amplitude but only phase.

To estimate the possible maximum data-rate for given channel and given noise, we can use Q-function as shown in Fig. 2.7 [20] . For Gaussian random variable Y with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma^2}$, Q-function is defined as follows:

$$\boldsymbol{Q(x) = P(Y > y) = P(X > x)}$$

$$\boldsymbol{= \frac{1}{\sqrt{2\pi}} \int_x^\infty exp\left(-\frac{u^2}{2}\right) du}, \qquad (2.14)$$

where

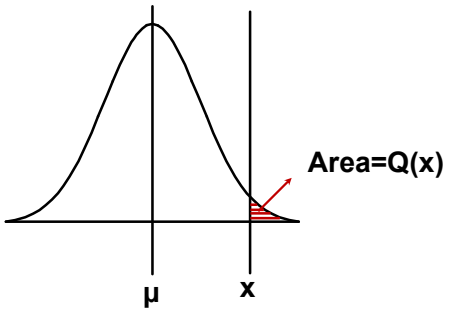


Fig. 2.7 Definition of Q-function.

$$\boldsymbol{x = \frac{y-\mu}{\sigma}} \ . \qquad (2.15)$$

Fig. 2.8(a) shows two Gaussian noise centered on the two worst case points. BER can be written as

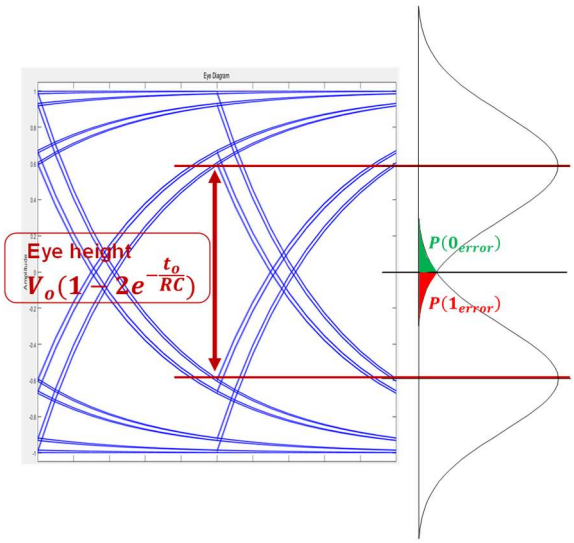
$$\boldsymbol{BER = P(error) = 0.5P(1_{error}) + 0.5P(0_{error})}$$

$$\boldsymbol{= Q\left(\frac{V_{eyeheight}/2}{\sigma}\right) = Q\left(\frac{V_o(1-2e^{-\frac{t_o}{RC}})}{2\sigma}\right)} \ . \qquad (2.16)$$

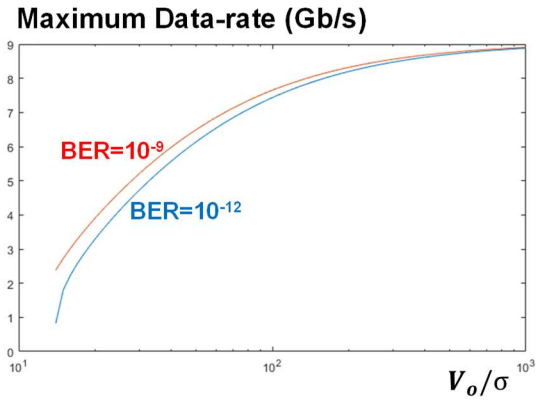
So, we can calculate the maximum data-rate with channel characteristic, $\boldsymbol{\sigma}$ of noise and target BER as follows:

$$\boldsymbol{1UI = t_o = -RCln\left(\frac{1}{2} - \frac{\sigma}{V_o}Q^{-1}(BER)\right)} \ . \qquad (2.17)$$

Fig. 2.8(b) shows the simulated results when the target BER is 10^{-9} or 10^{-12}.



(a)



(b)

Fig. 2.8 (a) BER estimation with two Gaussian noise centered on the two worst case points. (b) Estimation of maximum data-rate with given channel and noise characteristics.

2.1.2 Equalizer

2.1.2.1 CTLE

To fully compensate for channel loss, the transfer function of the equalizer should be inverse of the channel transfer function as shown in Fig. 2.9. When two functions are multiplied, a flat response is obtained. However, when noise is present and equalizer is not band-limited, noise can be more boosted than signal.

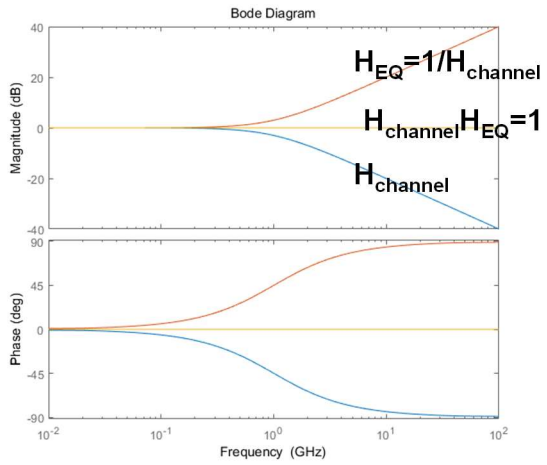


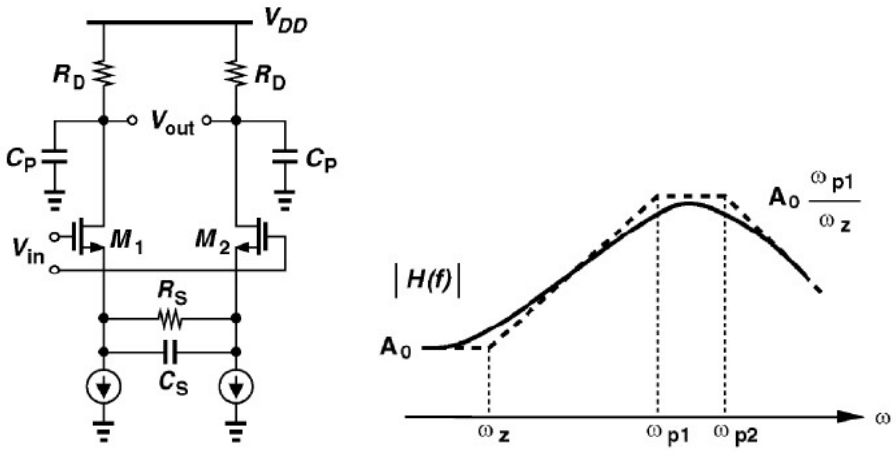
Fig. 2.9 Linear equalizer whose transfer function is inverse of the channel.

Fig. 2.10 shows general circuit and frequency response of continuous time linear equalizer (CTLE) [2] . With resistive and capacitive source degeneration R_s and C_s, the frequency response is obtained as follows:

$$\boldsymbol{H(s)} = \frac{g_m R_D}{\left(1+\frac{g_m R_S}{2}\right)} \frac{\left(1+\frac{s}{w_z}\right)}{\left(1+\frac{s}{w_{p1}}\right)\left(1+\frac{s}{w_{p2}}\right)} \tag{2.18}$$

where

$$\boldsymbol{w_z} = \frac{1}{R_S C_S} \ , \boldsymbol{w_{p1}} = \frac{1+\frac{g_m R_S}{2}}{R_S C_S} \ , \boldsymbol{w_{p2}} = \frac{1}{R_D C_P} \quad . \tag{2.19}$$



(a) (b)

Fig. 2.10 (a) Circuit and (b) frequency response of CTLE [2] .

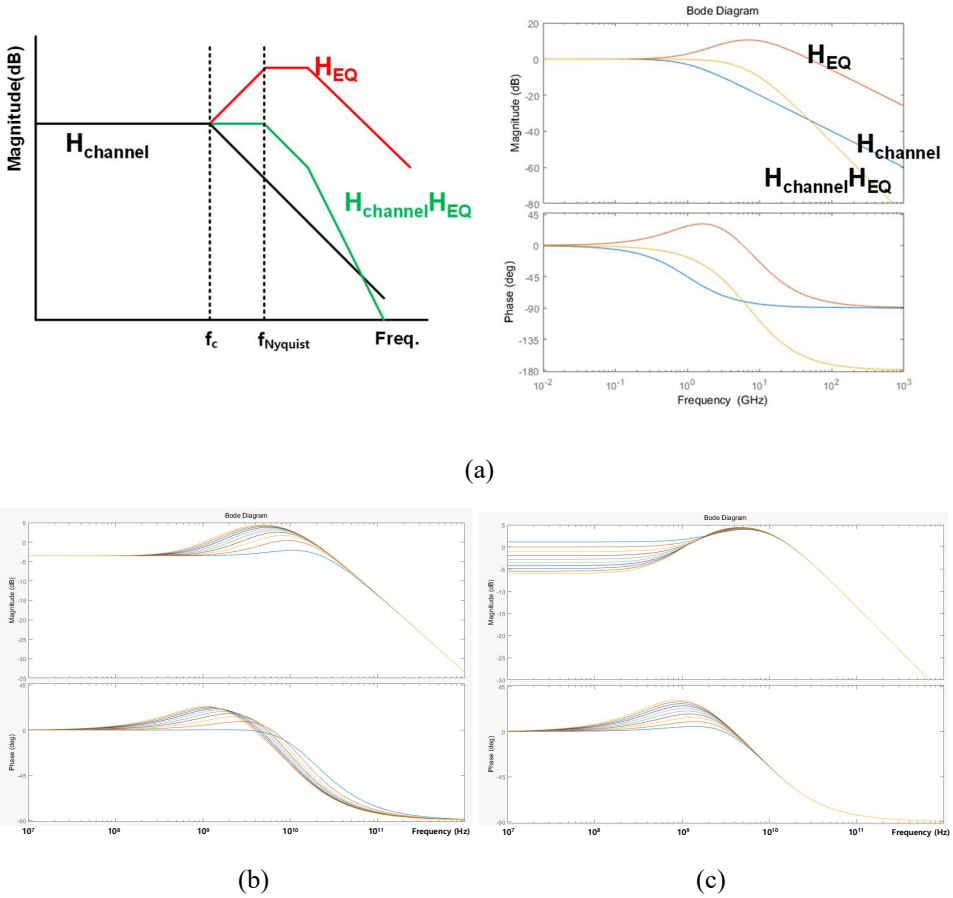


Fig. 2.11 (a) Compensation of channel with CTLE. Frequency response when (b) C_s and (c) R_s are controlled.

With this characteristics, the loss of channel is compensated as shown in Fig. 2.11(a). When f_z and f_{p1} are set to f_c of channel and $f_{Nyquist}$ respectively, the multiplied result shows almost flat response until $f_{Nyquist}$. Fig. 2.11(b) and (c) show frequency response when C_s and R_s are controlled, respectively.

However, although the bandwidth is limited with second pole, excessive boosting by CTLE cause noise boosting. Fig. 2.12 shows three sources of noise. Noise injected before or through channel can be fully or partially attenuated by channel response. However, noise injected after channel and before CTLE cannot be attenuated, and can be the main source of noise boosting by CTLE.

For example, as shown in Fig. 2.13, let's assume two channels CH1 and CH2 whose corner frequency is at 1GHz and 4GHz, respectively. For 8Gb/s data signal transmission, CH1 requires CTLE with f_z and f_{p1} at 1GHz and 4GHz, respectively. For CH2, only channel output is observed without CTLE. With these settings, white

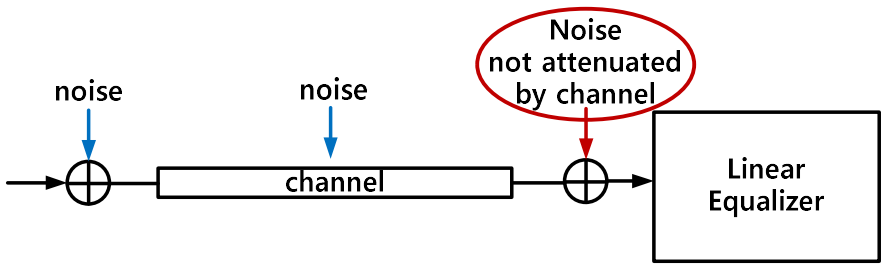


Fig. 2.12 Noise sources before linear equalizer.

Gaussian noise is injected before and after channel. The resulting eye diagrams are shown in Fig. 2.14. Without noise, the output of CH1 and CTLE and output of CH2 are similar because the loss of the channel is almost compensated by CTLE. Also, with noise before channel, they are similar because the noise canceled by attenuation and boosting. However, when the noise is injected after channel, noise itself is boosted by CTLE. As a result, the output of CH1 and CTLE is much degraded than the output of CH2.

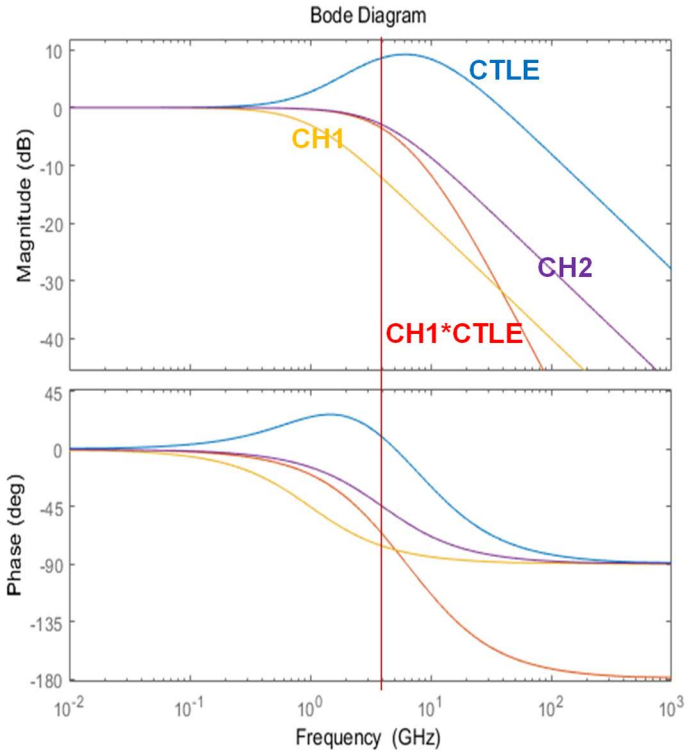


Fig. 2.13 Two channel example for noise boosting simulation.

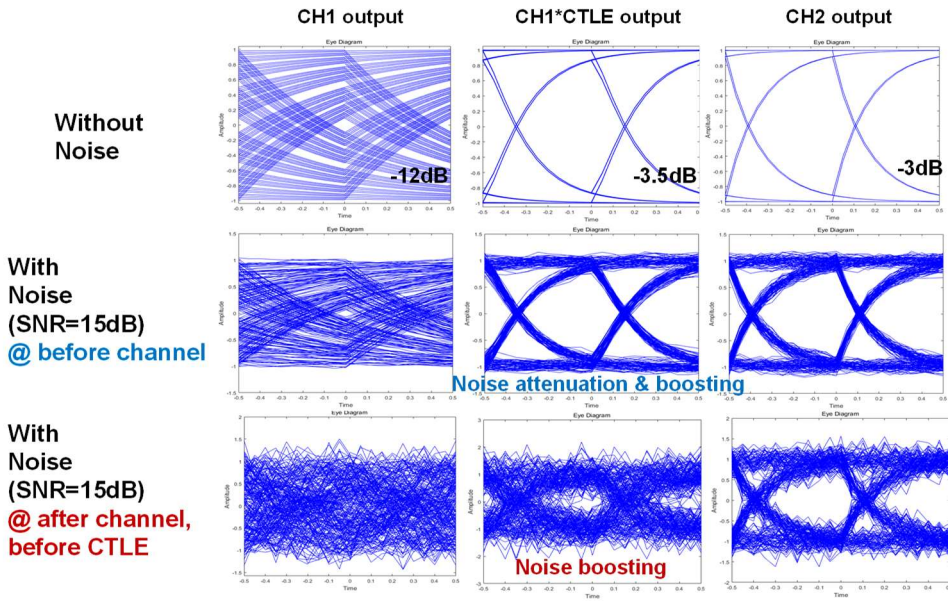


Fig. 2.14 Eye diagrams for CH1 output, CH1&CTLE output, CH2 output, respectively.

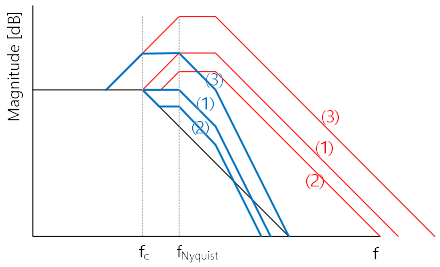
From the previous analysis, it is concluded that CTLE boosting should not be used excessively.

With f_z control with fixed f_{p1} and f_{p2} as shown in Fig. 2.15(a), signal ISI is minimized when the channel loss is perfectly canceled. ISI is increased when not only the CTLE boosting is less but also more than optimal amount. The power of AWGN after CTLE can be written as follows:

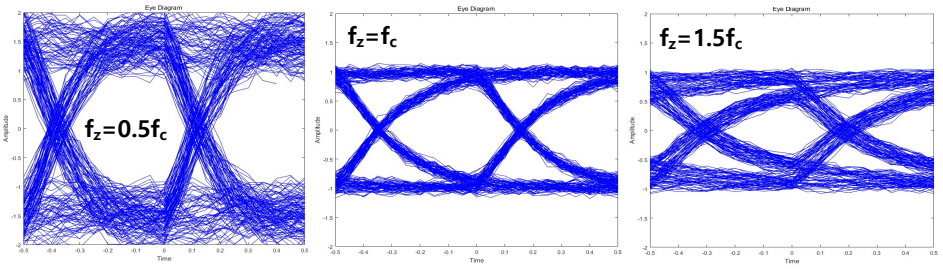
$$\boldsymbol{P_{noise} = \int_{-\infty}^{\infty} \frac{N_o}{2} \left| H_{EQ}(f) \right|^2 df} \tag{2.20}$$

where $N_o/2$ represents the power spectral density of AWGN. The noise power decreases as boosting decreases.

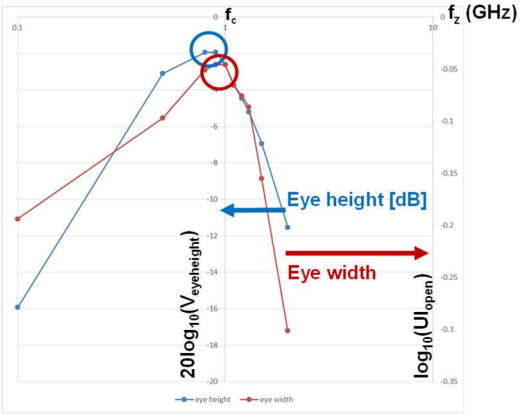
The simulated results of the combined effects of signal ISI and noise power with an SNR of 30dB are shown in Fig. 2.15(b) and (c). Eye height and width are maximized when the zero frequency is located near the corner frequency of the channel, to cancel the channel loss and not to boost noise too much.



(a)



(b)



(c)

Fig. 2.15 CTLE simulation with SNR=30dB. (a) f_z control in CTLE. (b) Eye diagrams for various f_z. (c) Eye height and Eye width according to f_z.

2.1.2.2 DFE

The architecture of decision feedback equalizer is shown in Fig. 2.16(a) [21] - [22] . By subtracting the result of being sampled, delayed and multiplied by coefficients w_n from input signal, the post-cursor ISI can be canceled. The SBR before and after DFE summer are shown in Fig. 2.16(b). Since we cannot predict the future value, the pre-cursor ISI cannot removed from DFE.

With illustration shown in Fig. 2.17(a), the signal expressions for node A, B and C for 2-tap DFE are as follows:

$$\boldsymbol{V_{nodeA} = V_m h_0 x[k] + V_m h_1 x[k-1] + V_m h_2 x[k-2]} \ , \tag{2.21}$$

$$\boldsymbol{V_{nodeB} = V_m w_1 x[k-1] + V_m w_2 x[k-2]} \tag{2.22}$$

and

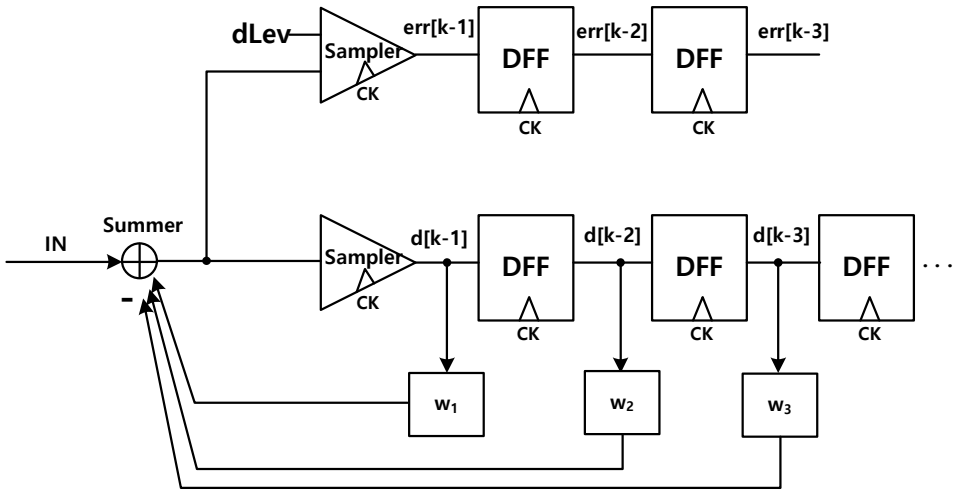
$$\boldsymbol{V_{nodeC} = V_m h_0 x[k]} \tag{2.23}$$

when h_n=w_n . Then, the transfer function for DFE can be written as follows:

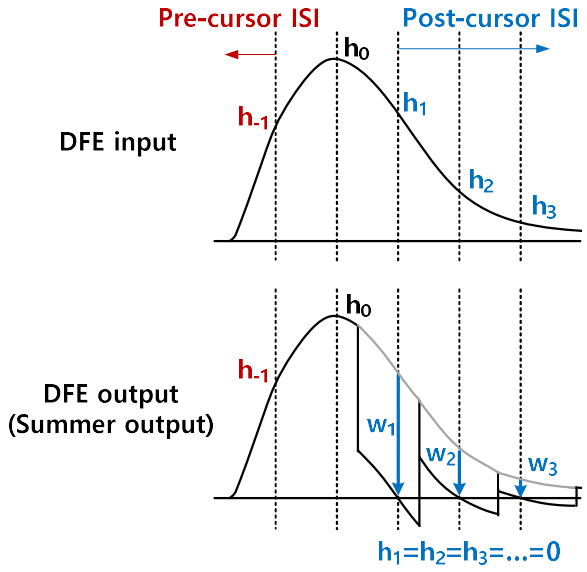
$$\boldsymbol{H = \frac{V_{nodeD}}{V_{nodeA}} = \frac{x[k]}{V_m h_0 x[k] + V_m h_1 x[k-1] + V_m h_2 x[k-2]}} \tag{2.24}$$

and it can be rewritten as

$$\boldsymbol{H(z) = \frac{1}{V_m h_0 + V_m h_1 z^{-1} + V_m h_2 z^{-2}} = \frac{1}{V_m} \frac{1}{h_0 + h_1 z^{-1} + h_2 z^{-2}}} \ . \tag{2.25}$$

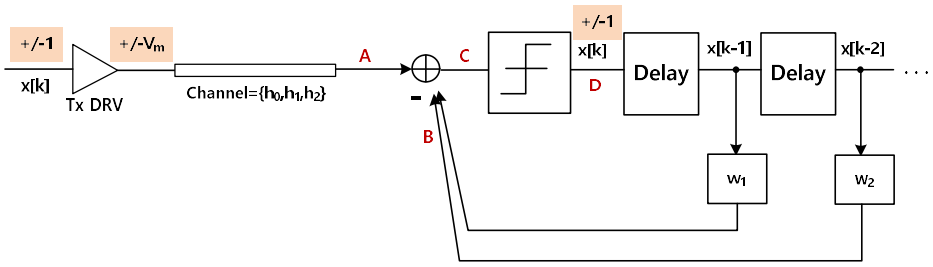


(a)

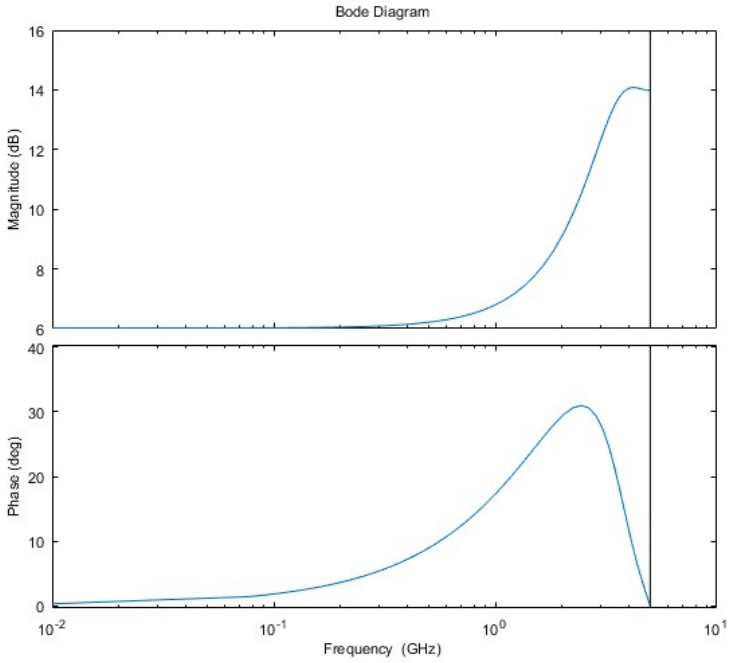


(b)

Fig. 2.16 (a) Structure of DFE. (b) SBR before and after DFE.



(a)



(b)

Fig. 2.17 (a) DFE representation to obtain transfer function. (b) Frequency response of DFE.

The frequency response from transfer function is also shown in Fig. 2.17(b). For example, V_m=0.5, h_0=0.6, h_1=w_1=0.3, h_2=w_2=0.1, 10Gb/s data are used. Since the decided digital values are used to cancel the ISI components, noise from input and DFF output are uncorrelated. So, DFE does not boost noise. So, generally, DFE is mainly used to cancel post-cursor ISI components without noise boosting and CTLE with moderate boosting is suitable for tail ISI cancellation [11] .

As shown in Fig. 2.18(a) and (b), the decision of the previous bits must be returned in time for sampling the current bit for right operation. Especially, the constraint on the first tap is important, and this value determines the maximum operating speed of the DFE. To release this constraint, loop-unrolling scheme can be used as shown in Fig. 18(c) [23] , [24] . After pre-calculation is done for all the cases that the decided value is +1 or -1, final result is selected by multiplexer according to the currently decided value.

Interleaved DFE such as half or quarter-rate DFE can be used as shown in Fig. 2.19 [25] . Although the hardware complexity and load seen from input node is increased and there are another implementation issues such as skew matching between multi-phase clocks, the operating speed of the circuits from the sampler can be released and the overall circuit design becomes efficient. However, the feedback delay constraint still remains.

The coefficients for DFE can be determined by least mean square algorithm as follows [21] :

$$\boldsymbol{w_n[k+1] = w_n[k] + \mu_w err[k] d[k-n]} \tag{2.26}$$

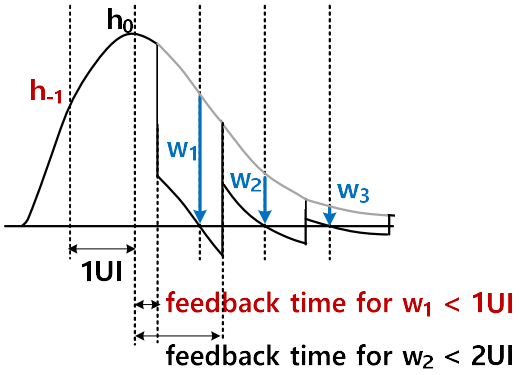
and

$$\boldsymbol{dLev[k+1] = dLev[k] + \mu_{dlev} err[k] d[k]} \ , \tag{2.27}$$

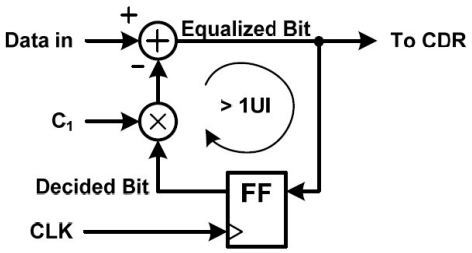
where

$$\boldsymbol{err[k] = (DFE\ \ summer\ \ out) - dLev[k]} \ . \tag{2.28}$$

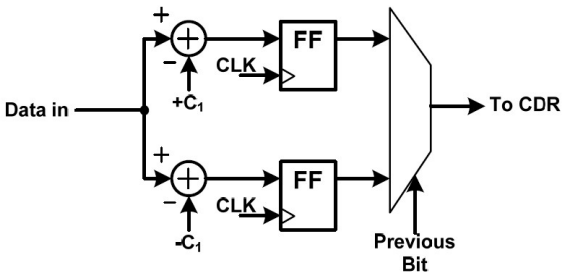
Variables k, μ and n mean each adaptation step, adaptation speed and n^{th} tap, respectively. This algorithm converges to the state where the mean square error is minimized. More details will be covered in Chapter3.



(a)



(b)



(c)

Fig. 2.18 (a) Feedback constraint. (b) Direct feedback DFE. (c) Loop unrolling DFE [24] .

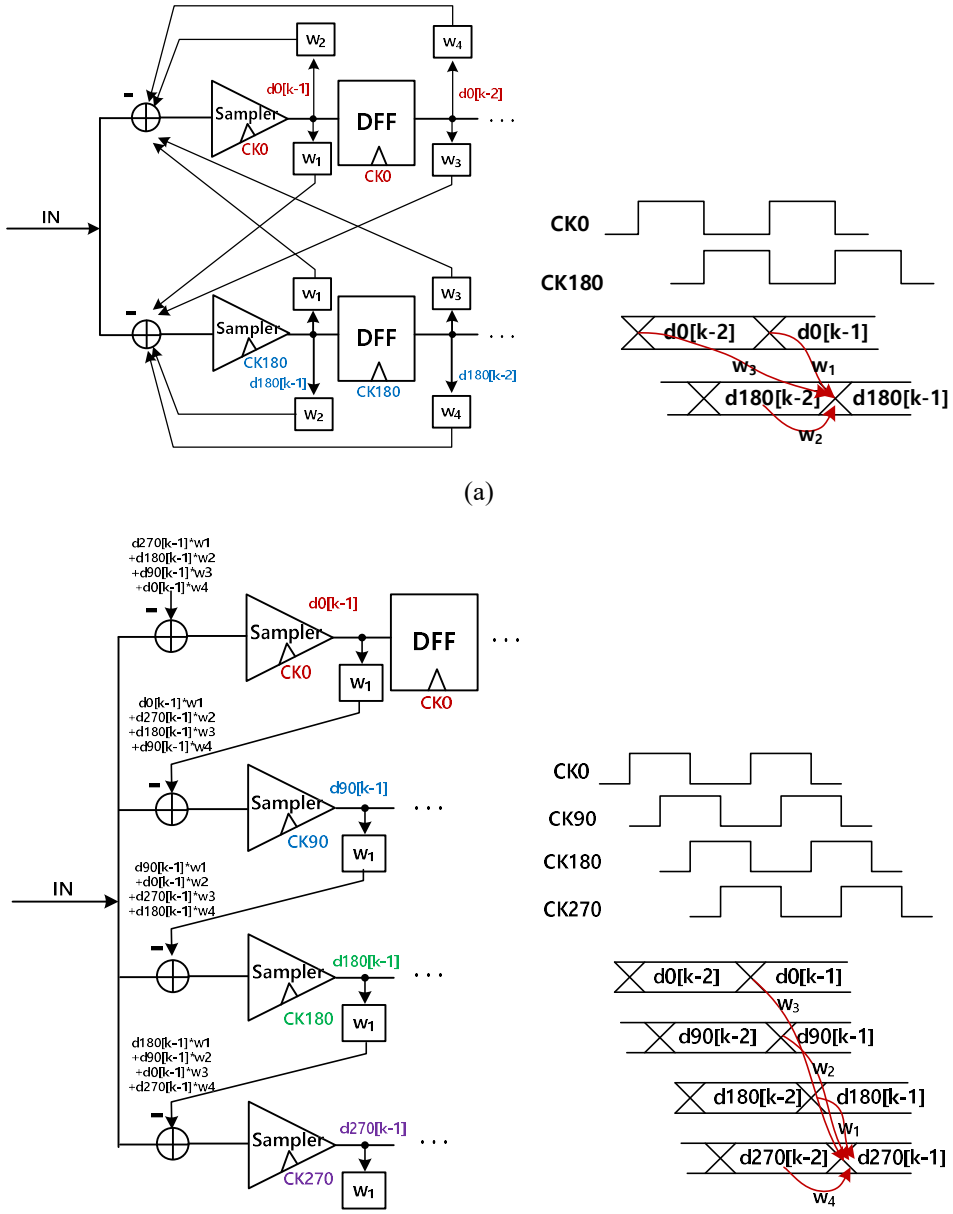


Fig. 2.19 (a) Half rate DFE. (b) Quarter rate DFE.

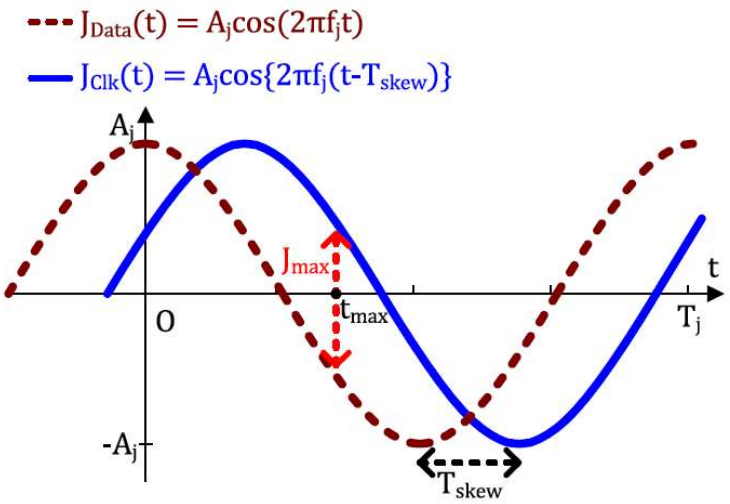
2.1.3 CDR

In serial communication of digital data, clock and data recovery is the process of extracting timing information and decoding the transmitted symbols [26] , [27] . When the transmitter does not transmit the clock signal along with the data stream, the clock should be generated at the receiver, using the timing information from the data stream. When there is a channel lane for clock, there is a significant reduction in power consumption and area required for the timing recovery circuits [28] . Since the proposed architecture in this study adopts the forwarded clock architecture, jitter analysis for the forwarded clock architecture is described in this chapter.

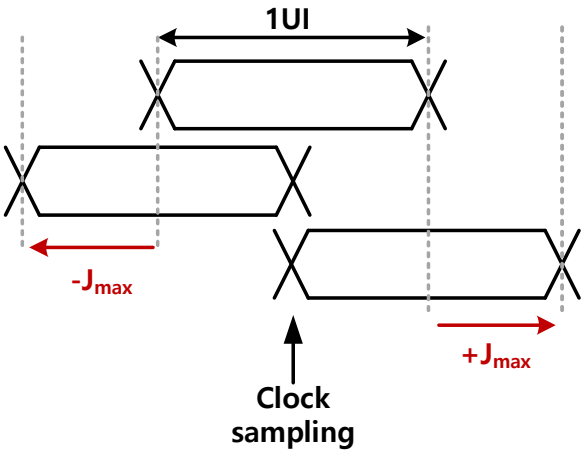
2.1.3.1 Jitter Characteristics of Forwarded Clock Architecture

Jitter tolerance is the peak-to-peak amplitude of sinusoidal jitter applied on the data input that causes the target BER threshold [29] , [30] . It is one of the indicators of CDR performance.

Fig. 2.20(a) shows jitter profiles of the forwarded clock and the received data, assuming that fully correlated sinusoidal jitter is contained in the clock and data. [28] . With timing skew T_{skew} between data and clock, the timing error can be written as follows:



(a)



(b)

Fig. 2.20 (a) Sinusoidal jitter profiles of the forwarded clock and the received data [28] .

(b) Relation between data and clock sampling phase.

$$|J_{data}(t) - J_{clk}(t)| = \left|A_j \cos(2\pi f_j t) - A_j \cos\{2\pi f_j (t - T_{skew})\}\right| = |2A_j \sin\left\{2\pi f_j \left(t - \frac{T_{skew}}{2}\right)\right\} \sin(\pi f_j T_{skew})|. \quad (2.29)$$

Then t_{max}, the moment when the error is maximized, can be expressed as

$$t_{max} = \frac{1}{2}\left(T_{skew} + \frac{T_j}{2}\right) . \quad (2.30)$$

With t_{max} and relationship shown in Fig. 2.20(b), the maximized timing error can be expressed as

$$J_{max} = |2A_j \sin(\pi f_j T_{skew})| < 0.5\text{UI} . \quad (2.31)$$

Therefore, the maximum peak-to-peak sinusoidal jitter boundary that corresponds to the jitter tolerance becomes

$$J_{pp} = 2A_j < \frac{0.5UI}{\sin(\pi f_j T_{skew})} . \quad (2.32)$$

The simulated jitter tolerance from (2.32) with T_{skew} of 1ns is shown in Fig. 2.21. The corner frequency where the JTOL become 1UI can be obtained as follows:

$$\frac{0.5UI}{\sin(\pi f_{j,corner} T_{skew})} = 1UI \quad (2.33)$$

and

$$f_{j,corner} = \frac{1}{6T_{skew}} . \quad (2.34)$$

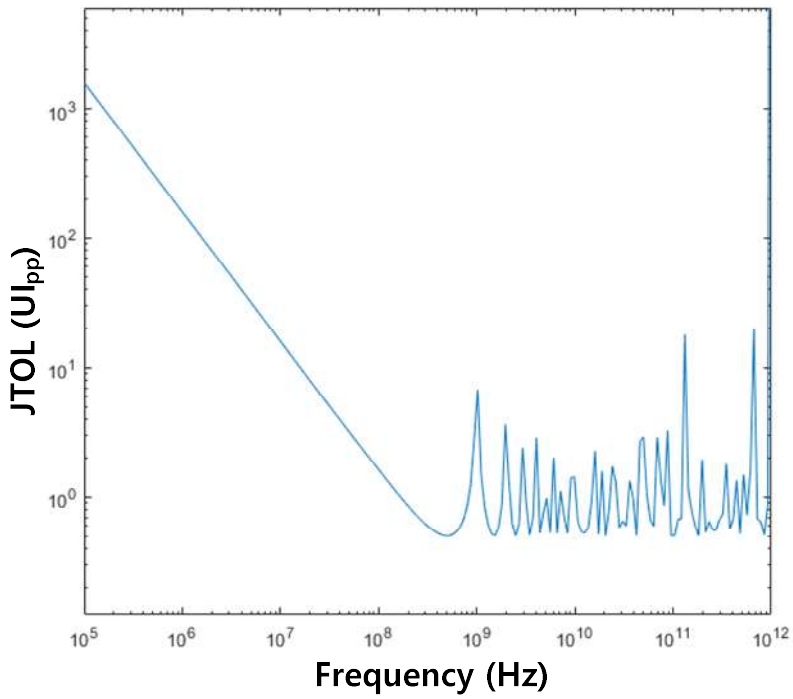


Fig. 2.21 Simulated jitter tolerance from (2.32).

The jitter tolerance of the forwarded clock architecture can be obtained in another way using jitter transfer function [31] . First, let's analyze the jitter transfer function of DLL. As shown in Fig. 2.22, output of the voltage controlled delay line (VCDL) in DLL is a delayed version of the input clock. The relations between input, output and error phases can be written as follows:

$$\boldsymbol{\phi_{err} = \phi_{in} - \phi_{out} e^{-sT}} \tag{2.35}$$

and

$$\boldsymbol{\phi_{out} = \phi_{in} + \phi_{err} \frac{K}{s} \frac{2\pi}{T} = \phi_{in} + \left(\phi_{in} - \phi_{out} e^{-sT}\right) \frac{K}{s} \frac{2\pi}{T}} \tag{2.36}$$

where K and 1/s mean a variable to convert phase error information to delay amount and integration, respectively. Then the jitter transfer function can be expressed as

$$\boldsymbol{H(s) = \frac{\phi_{out}}{\phi_{in}} = \frac{sT + 2\pi K}{sT + 2\pi K e^{-sT}}} \ . \tag{2.37}$$

As well known, the jitter transfer function is close to all-pass filter [32] .

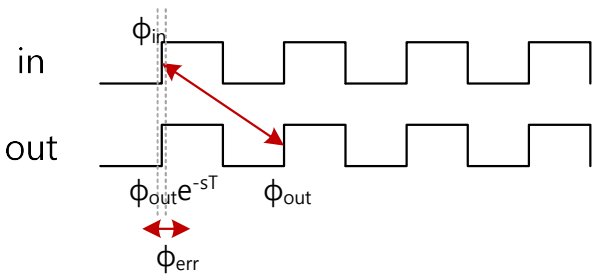


Fig. 2.22 Input and output clocks in DLL.

With similar approach, we can develop expressions for forwarded clocking architecture when T_{skew} between data and clock is α times of clock period as shown in Fig. 2.23.

$$\boldsymbol{\phi_{err} = \phi_d e^{(\alpha-1)sT} - \phi_c e^{-sT}} \tag{2.38}$$

and

$$\boldsymbol{\phi_c = \phi_d e^{(\alpha-1)sT} + \phi_{err} \frac{K}{s} \frac{2\pi}{T}}$$

$$\boldsymbol{= \phi_d e^{(\alpha-1)sT} + \left(\phi_d e^{(\alpha-1)sT} - \phi_c e^{-sT}\right) \frac{K}{s} \frac{2\pi}{T}} \ . \tag{2.39}$$

Then the jitter transfer function becomes

$$\boldsymbol{H(s) = \frac{\phi_d}{\phi_c} = e^{(\alpha-1)sT} \frac{sT + 2\pi K}{sT + 2\pi K e^{-sT}}} \ . \tag{2.40}$$

With jitter transfer function, the jitter tolerance can be obtained as follows:

$$\boldsymbol{-\pi < \phi_{in}\left(1 - H(s)\right) < +\pi} \ , \tag{2.41}$$

$$\boldsymbol{\frac{-\pi}{1-H(s)} < \phi_{in} < \frac{+\pi}{1-H(s)}} \ , \tag{2.42}$$

then

$$\boldsymbol{J_{pp} < \frac{1UI}{1-H(s)}} \ . \tag{2.43}$$

The simulated error function and jitter tolerance function from (2.40) and (2.43) are shown in Fig. 2.24(a). In Fig. 2.24(b), two JTOL curves from (2.32) and (2.43)

are plotted together. In the frequency region of interest, two curves match well and they converge to about half-UI in the high frequency region.

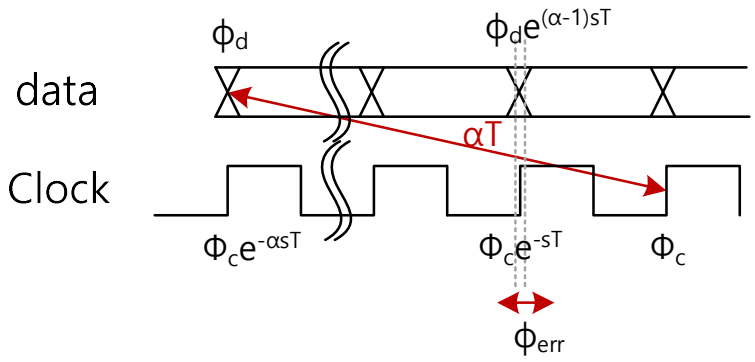
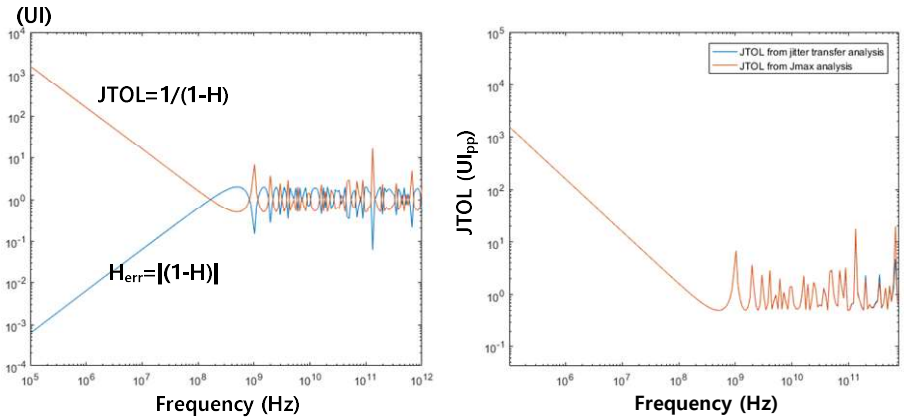


Fig. 2.23 Data and clock relation in forwarded clocking architecture when T_{skwe} is αT.



(a) (b)

Fig. 2.24 (a) Simulated H_{err} and JTOL from (2.40) and (2.43). (b) Comparison of JTOL curves between two different analyses from (2.32) and (2.43).

2.2 Prior Arts on Clock Recovery

2.2.1 BB-CDR

One of the most widely used CDRs is BB-CDR [33] -[35] . Using edge sample and data samples, the BB-PD generates early and late signals as shown in Fig. 2.25(a) and (b). After convergence, the edge sample is placed at the zero crossing of the data stream as shown in Fig. 2.25(c). The hardware and the operation of BB-CDR is simple and that is the reason why BB-CDR is widely used. However, when viewed on a single bit response, it converges at a phase where h(+0.5) and h(-0.5) are the same, as already mentioned in Chapter1 with Fig. 1.1. The lock phase may change according to the shape of single bit response, and that phase is not an optimal phase to minimize BER.

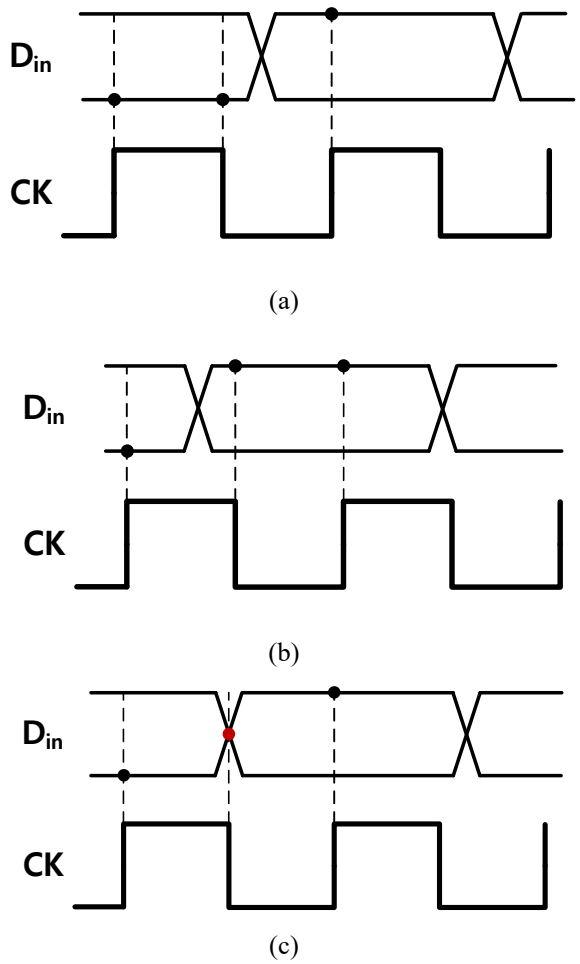


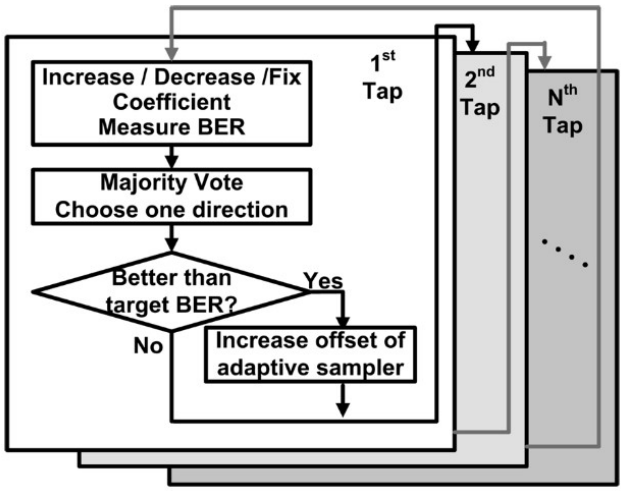
Fig. 2.25 Concept of BB-CDR. (a) When clock is early. (b) When clock is late. (c) The edge sample is placed at the zero crossing after lock.

2.2.2 BER-Based CDR

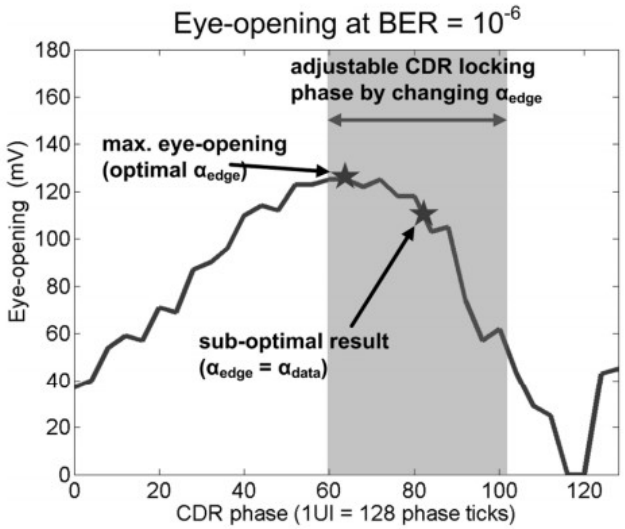
In order to find the optimal phase to minimize the BER, CDRs based on BER counting have been proposed.

Fig. 2.26(a) shows the basic steepest gradient algorithm [4] . Although this flow chart represents the search algorithm for the tap coefficient of the equalizer, the sampling phase search is also based on this algorithm. After moving the phase by one step, BER is measured with majority voting to reduce the effect of noise. When the current BER result is compared with the previous result, if the result is better, the current direction is maintained, and if it is worse, another direction is tried. Fig. 2.26(b) shows that the phases found with BB-CDR is different from the phase obtained through the minimum BER algorithm. For iterative operation and BER counting, hardware becomes complicated and long processing time is required. In particular, when the BER target is low, the processing time increases exponentially.

The concept of second BER-based CDR is shown in Fig. 2.27 [5] . The stochastic hill climbing algorithm is used instead of the basic steepest descent method. It measures the BER while randomly perturbating the variables to be adapted, including the sampling phase. Then the number of iterations and processing time are reduced compared to the steepest descent algorithm.

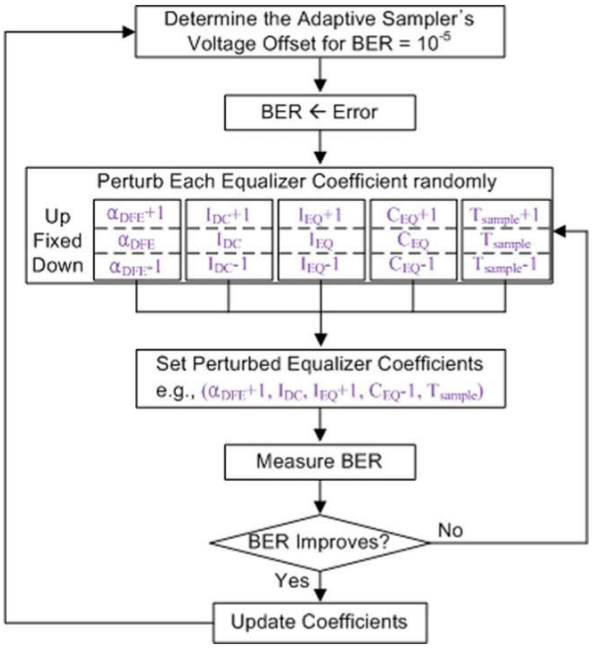


(a)

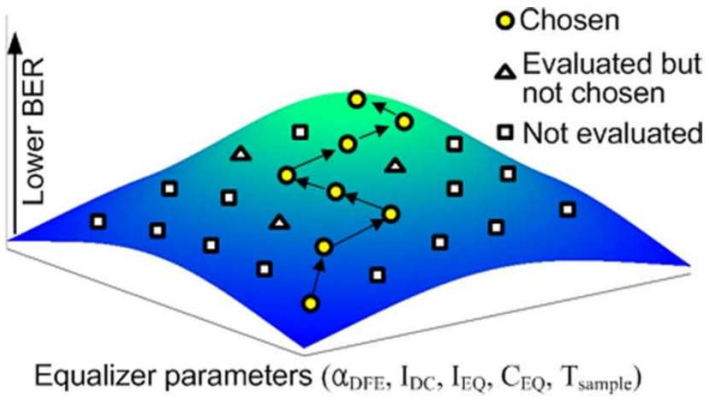


(b)

Fig. 2.26 (a) Steepest gradient algorithm and (b) eye opening according to CDR phase in [4] .



(a)



(b)

Fig. 2.27 (a) Flow chart and (b) concept of stochastic hill-climbing algorithm in [5] .

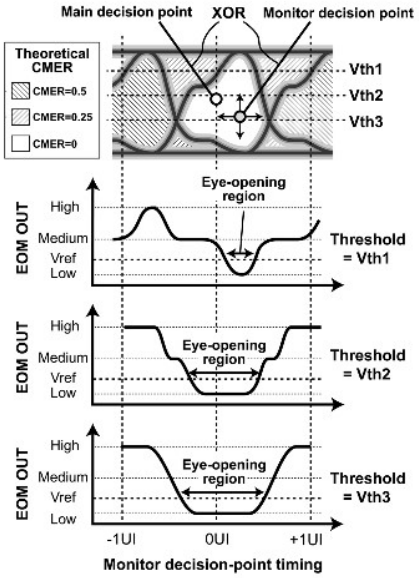
2.2.3 EOM-Based CDR

There have been CDRs that do not measure the BER directly, but use the eye opening monitor to find the optimal phase.

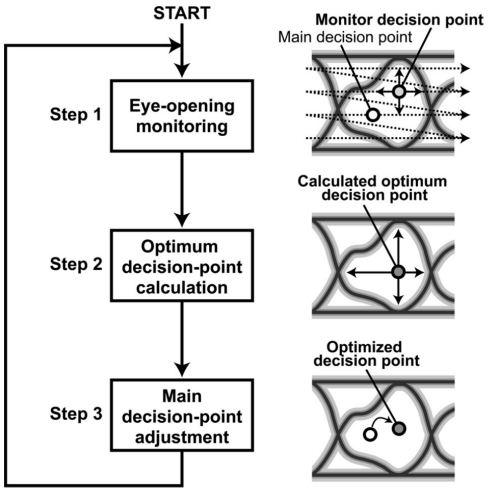
Fig. 2.28 shows a concept of CDR with EOM by defining CMER, or code mismatch error rate [6] . When the main decision point and the monitor decision point are inside the same eye region, CMER becomes 0. If they belong to different eyes, CMER becomes 0.5. The monitor decision point is determined by sweeping 128 steps each on the x and y axes. With a reference voltage between 0 and 0.25, it finds the approximate eye boundary. The decision point is obtained by finding the point where the x and y margins are maximized. Because it does not measure the exact BER, it can save time than BER-based CDRs.

Fig. 2.29 shows EOM-based CDR using the PDF and CDF information [7] . When the distributions of 1 and 0 in the eye diagram are expressed in Gaussian distribution function, the effective eye height is defined by the distance from the mean value to sigma. The phase where the effective eye height is maximized is found by sweeping 128 steps.

While the indirect criteria require a shorter processing time than the BER counting in finding the optimal phase, EOM-based CDRs still require complex hardware and long processing time because they operate iteratively by sweeping the sampling phase.

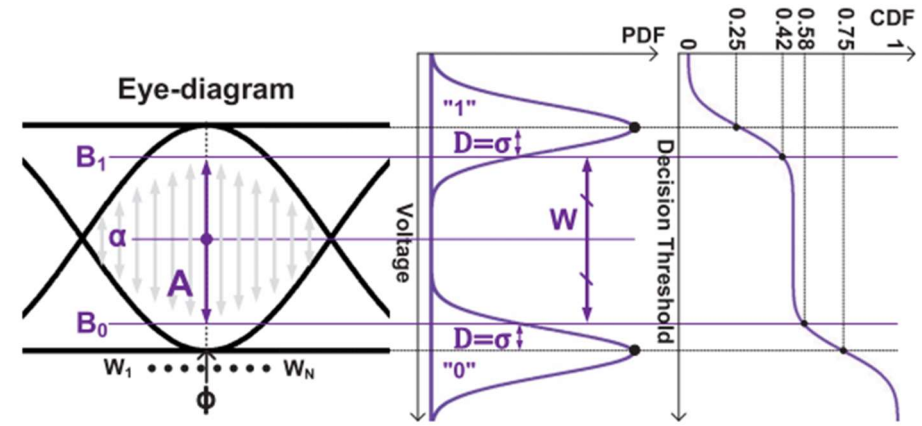


(a)

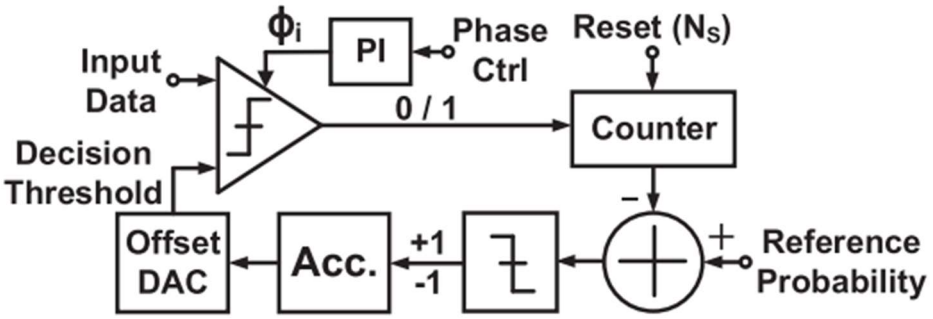


(b)

Fig. 2.28 (a) Definition of CMER and output of EOM. (b) Procedure of EOM in [6] .



(a)



(b)

Fig. 2.29 (a) Eye height definition from PDF and CDF and (b) architecture of [7] .

2.3 Concept of the Proposed CDR

To summarize the pros and cons of the previous works, BB-CDR is simple but cannot find the optimal phase and BER or EOM based CDRs can search for the optimal phase, but require complex hardware and long processing time. The purpose of the proposed CDR in this study is to take advantage of each architecture. With simple hardware like BB-CDR, the proposed CDR tracks near-optimal sampling phase and complete the adaptation in a short time.

The sampling phase for the maximum vertical eye margin, the maximum horizontal eye margin, and the minimum BER may not perfectly coincide, varying over the channel characteristics. In this work, the vertical eye margin is used as a criterion for near-optimal sampling phase. The basic concept is the same as the assumption in EOM-based CDRs: when the eye height is the maximum, the BER also approaches the minimum. Fig. 2.30(a) and (b) represent BER and vertical voltage margin of eye diagram according to equalizer coefficients, respectively [4] . The results of two graphs are almost similar. Using this feature, the propose CDR tracks the point where the eye height is maximum to minimize BER. Therefore, it is named maximum-eye-tracking CDR, MET-CDR for short [8] .

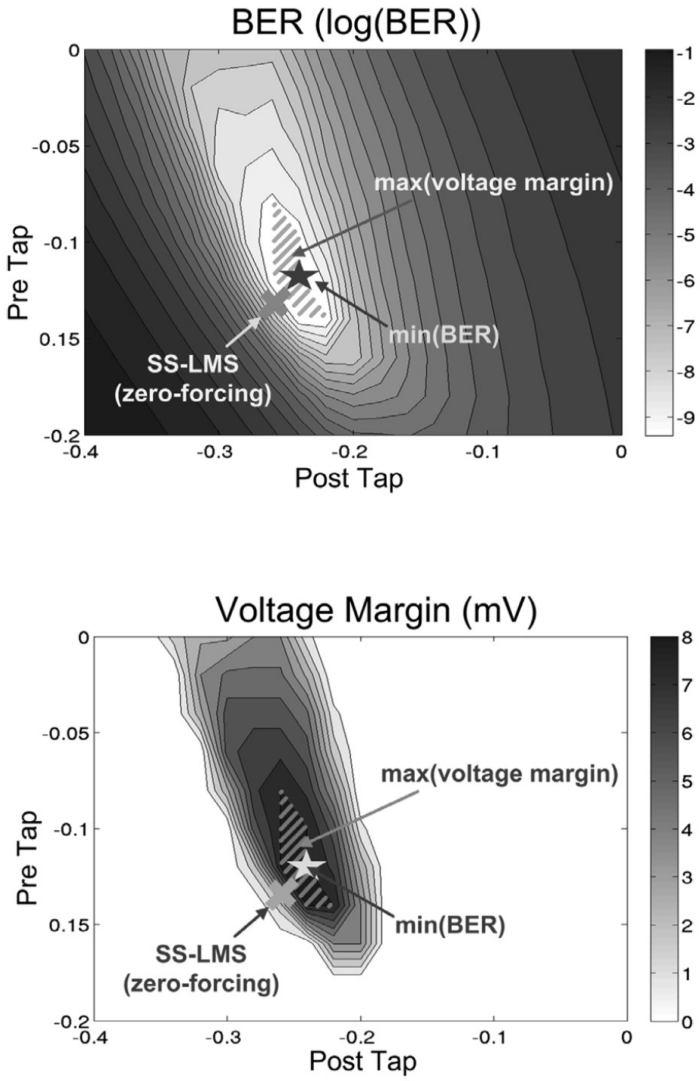


Fig. 2.30 Relation between BER and voltage margin in [4] .

Chapter 3

Maximum-Eye-Tracking CDR with Biased Data-Level and Eye Slope Detector

3.1 Overview

In this chapter, the design of the proposed maximum-eye-tracking CDR is presented. The concept of the biased data-level, eye slope detector and adaptation algorithm is explained and the operation is verified with MATLAB simulation.

Basically, the proposed architecture is based on NRZ signaling. At the end of the chapter, the expansion of the proposed MET-CDR to PAM4 signaling with simulation and future works are mentioned.

3.2 Design of MET-CDR

3.2.1 Eye height information from biased data-level

At the input of the sampler or the output of the summer in the decision feedback equalizer (DFE), the eye height can be defined as follows:

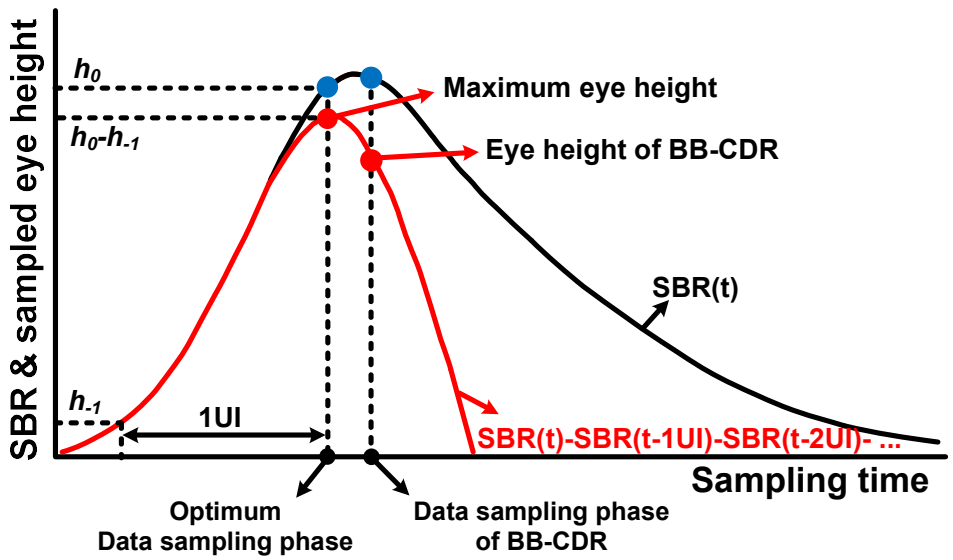
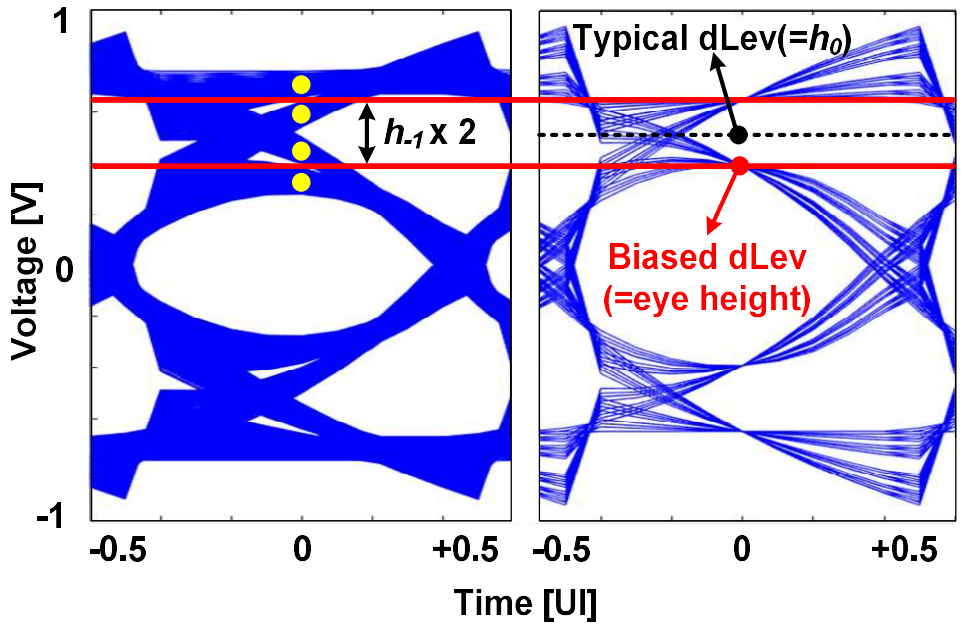


Fig. 3.1 Eye height (EH) calculated with main cursor and pre-cursor ISI and the optimum sampling phase determined by the maximum eye height.

$$\boldsymbol{EyeHeight = h_0 - \sum_{n=1}^{\infty} |h_n| - \sum_{n=-\infty}^{-1} |h_n|} \quad . \tag{3.1}$$

The second and the third terms represent post-cursor ISI and pre-cursor ISI, respectively. For simplicity, we assume that the DFE is ideal and the number of taps is large enough to cover all the post-cursors. Then we can ignore the second term in (3.1). However, the third term still remains because the DFE cannot remove the pre-cursor ISI. The continuous-time linear equalizer (CTLE) can sharpen the SBR, but cannot directly remove the pre-cursor ISI [11] . As a result, the eye height at the sampler input can be obtained by subtracting the sum of the pre-cursor ISI from the main-cursor. In Fig. 3.1, the two curves show the calculated eye height variations from (3.1) with the given SBR as a function of the sampling phase. The optimal sampling phase determined as the sampling phase of the maximum eye height appears earlier than the locked phase of the BB-CDR because the pre-cursor ISI is reduced faster than the main cursor as the sampling position is pulled forward [4] , [5] . Fig. 3.2(a) and (b) show the MATLAB-simulated eye diagrams during and after the DFE adaptation, respectively. The horizontal solid lines indicate two data-levels generated when one pre-cursor ISI is present. With a conventional LMS algorithm, the data-level (dLev) is obtained as follows:



(a) (b)

Fig. 3.2 Simulated eye diagram (a) during and (b) after the DFE adaptation with one pre-cursor ISI.

$$\boldsymbol{dLev[k+1] = dLev[k] + \mu_{dLev} \times err[k] \times d[k]} \tag{3.2}$$

where μ_{dLev}, err[k] and d[k] are step size, error value for dLev and sampled data, respectively. The resulting dLev is equal to h_0, the center of the data pattern 1 in the eye diagram. However, as shown in Fig. 3.2(b), the eye height is smaller than h_0 by h_{-1}. To account for the effect of the pre-cursor ISI, we define a 'biased dLev'. The term 'biased' means that when determining dLev, a weighted sum of UP and DN in the ratio of 1: α is used in the sign-sign LMS (SS-LMS) algorithm as follows:

$$\begin{aligned} &\boldsymbol{if\ \ err[k] > 0}\ \textbf{ (dLev 'UP'):} \\ &\quad \boldsymbol{dLev[k+1] = dLev[k] + 1 \times \mu_{dLev} \times sign(err[k] \times d[k])} \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} &\boldsymbol{if\ \ err[k] < 0}\ \textbf{ (dLev 'DN'):} \\ &\boldsymbol{dLev[k+1] = dLev[k] + \alpha \times \mu_{dLev} \times sign(err[k] \times d[k])}\ . \end{aligned} \tag{3.4}$$

Assuming that the data pattern is random, the residual ISI errors represented by four dots in Fig. 3.2(a) contain the same number of hits. Therefore, the level of the lower line, or biased dLev, can be obtained by adding UP and DN with the weight ratio of 1:3. If there is no residual ISI error or noise after DFE adaptation as shown in Fig. 3.2(b), the biased dLev is equal to the eye height. In a similar approach, with two

pre-cursors, four lines divide the residual error equally into eight areas. Therefore, the desired weighting factor that achieves the lowest level is 1:7. For N pre-cursors, the weighting factor becomes $1:2^{N+1}-1$.

Fig. 3.3(a) shows data levels with additive white Gaussian noise (AWGN) assuming one pre-cursor ISI. Two distributions centered at h_0+h_{-1} and h_0-h_{-1} overlap each other. 1:3-*dLev* converges to the value that satisfies the below equation (5) where *A* and *B* represent the probability indicated by the diagonal patterns in Fig. 4(a):

$$(\mathbf{1}-\boldsymbol{B})+\left(\frac{\mathbf{1}}{\mathbf{2}}+\boldsymbol{A}\right):\boldsymbol{B}+\left(\frac{\mathbf{1}}{\mathbf{2}}-\boldsymbol{A}\right)=\mathbf{3}:\mathbf{1}\ . \tag{3.5}$$

and this equation can be simplified as $A=B$, implying that Δd is determined on the condition that the probability '*A*' corresponding to the area from the mean to Δd of the Gaussian distribution and the probability '*B*' corresponding to the area from $2h_{-1}+\Delta d$ away from the mean value to infinity are the same. As a result, 1:3-*dLev* is lowered from h_0-h_{-1} by Δd. Fig. 3.3.(b) shows the calculated Δd according to the relationship between h_{-1} and σ. When h_{-1} is larger than 1.5σ, Δd becomes almost 0 since about 99.7% of cases lie within $\pm 3\sigma$.

Note that it is important to check whether the sampling phase where biased *dLev* becomes maximum (t_0 in Fig. 3.4) remains the same in the presence of AWGN. The two eye height functions with and without Gaussian noise can be written as follows:

$$EH_{wo.noise}(t) - \Delta d(t) = EH_{w.noise}(t). \tag{3.6}$$

Differentiating (3.6) with respect to *t* is as below:

$$\frac{\partial EH_{wo.noise}}{\partial t} - \frac{\partial \Delta d}{\partial t} = \frac{\partial EH_{w.noise}}{\partial t} \ . \tag{3.7}$$

For the region before the peak ($t<t_0$),

$$\frac{\partial EH_{wo.noise}}{\partial t} > 0, \qquad \frac{\partial \Delta d}{\partial t} \leq 0 \ . \tag{3.8}$$

From (3.7) and (3.8),

$$\frac{\partial EH_{w.noise}}{\partial t} \geq \frac{\partial EH_{wo.noise}}{\partial t} \ . \tag{3.9}$$

From (3.9), we can predict that the slope of *EH* function with Gaussian noise is sharper than that without noise.

For the region after the peak ($t>t_0$),

$$\frac{\partial EH_{wo.noise}}{\partial t} < 0, \qquad \frac{\partial \Delta d}{\partial t} \leq 0 \ . \tag{3.10}$$

In order for the peak position not to change due to Δd, the inequality below should be satisfied:

$$\frac{\partial EH_{w.noise}}{\partial t} < 0 \ . \tag{3.11}$$

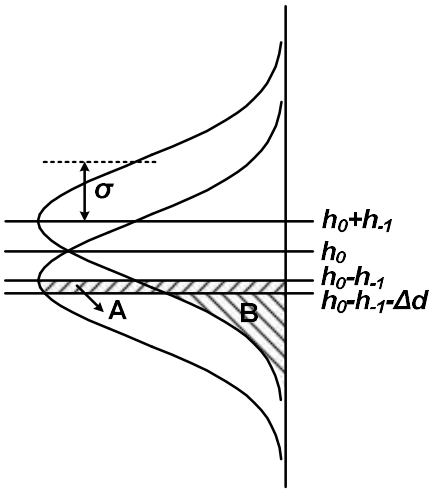
Then, (3.7), (3.10) and (3.11) lead to the condition

$$\left|\frac{\partial EH_{wo.noise}}{\partial t}\right| > \left|\frac{\partial \Delta d}{\partial t}\right| \ . \tag{3.12}$$

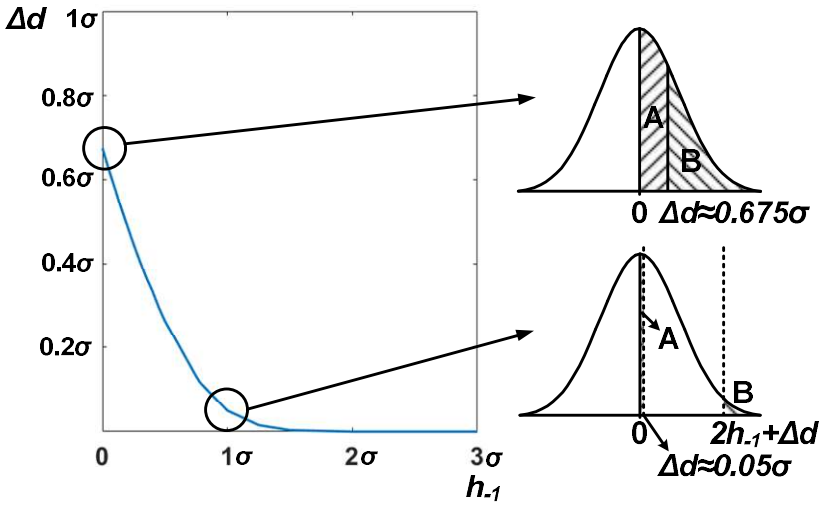
Fig. 3.4 shows the simulated results of $EH_{wo.noise}$ and $EH_{w.noise}$ with two channels whose losses at Nyquist frequency are about 10 dB and 20 dB, respectively. The peak value of the eye height should be larger than 7σ to achieve the target BER of 10^{-12}, so σ is set to 0.07 and 0.035 for two channels. As h_{-1} at the lock point increases, the absolute value of the slope of Δd decreases quickly as shown in Fig. 3.3(b). Under this circumstance, (3.12) can be met easily, and the peak position is not affected by noise.

The actual eye height is always smaller than the biased *dLev* because of the random noise, jitter from power supply noise and device noise. Since the amount of noise is the same regardless of the sampling phase, the actual eye height can be obtained by shifting down h_0-h_{-1} vertically, which does not change the peak position. In Fig. 3.4, the actual eye height h_0-h_{-1}-$N\sigma$ is shown. N varies depending on the target BER, and this example is when N=6 with target BER of 10^{-9}. To conclude, finding the maximum of the biased *dLev* indirectly leads to the sampling phase at the peak of the actual eye height in both presence and absence of noise. Hence, we can use the biased *dLev* as a criterion for finding the maximum eye

height. Simply changing the weighting factor from 1:1 to 1: α defines a meaningful level that can be used for optimal phase search.

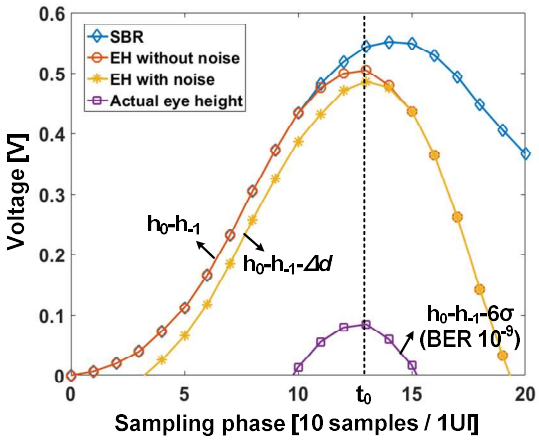


(a)

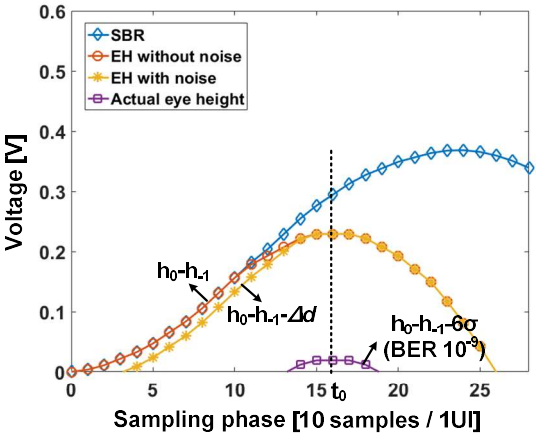


(b)

Fig. 3.3 (a) Biased dLev is lowered from h_0-h_{-1} by Δd when white Gaussian noise exists. (b) Calculated Δd according to the relationship between h_{-1} and σ.



(a)



(b)

Fig. 3.4.Simulated results of $EH_{wo.noise}$, $EH_{w.noise}$ and actual eye height for two channels. (a) 10 dB loss and σ=0.07. (b) 20 dB loss and σ=0.035.

3.2.2 Eye Slope Detector and Adaptation Algorithm

The eye height from (3.1) considering only pre-cursor ISI with sufficient DFE taps can be rewritten in time domain as follows:

$$EH(t) = SBR(t) - \sum_{n=-\infty}^{-1} |SBR(t + nT_s)| \tag{3.13}$$

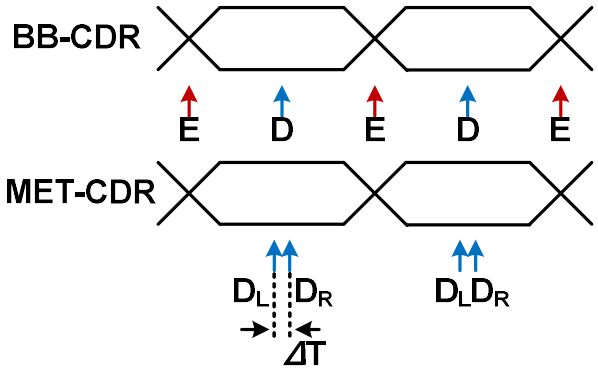
where T_s is one unit interval (UI). Since *EH(t)* is concave around the peak as shown in Fig. 3.1, the eye height can be maximized by finding the position of the main cursor, T_m, that satisfies the following equation [49]:

$$\left.\frac{\partial EH}{\partial t}\right|_{t=T_m} = 0. \tag{3.14}$$

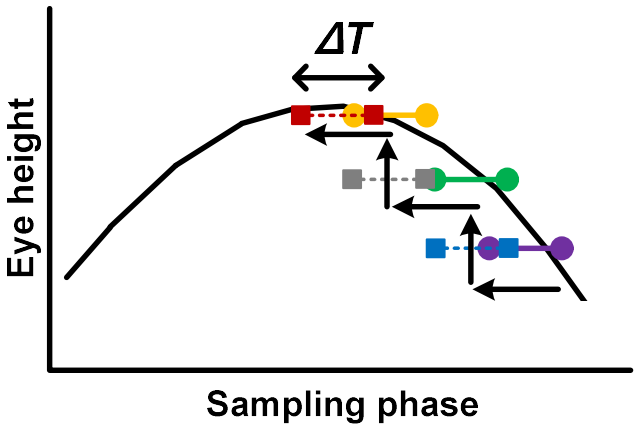
As shown in Fig.3.5(a), two samples on two slightly different timings are used in the MET-CDR to find the derivative value of EH. No edge samples are used. Fig. 3.5(b) shows one example of the convergence process according to the adaptation algorithm shown in Table 3.1. When the polarities of the left and right errors (L-error and R-error) are the same, the derivative of EH at current point is zero, so biased dLev and the DFE coefficients w_n are updated. When the signs are different from each other, the sampling phase is updated in the direction that EH increases to reach the point where the derivative value is zero. By repeatedly updating the sampling phase and dLev, two error samples detect both the eye height of current position and

the slope of the eye height. Eventually it converges on the maximum eye height where the slope becomes zero. To avoid interaction between the adaptive DFE loop and CDR leading to instability, the DFE loop works faster than the CDR loop. With low CDR loop bandwidth, the jitter tolerance is improved by adopting a forwarded clocking architecture [28] whose jitter tolerance bandwidth is a function of timing skew between the data path and the clock path.

Fig. 3.6(a) shows the simulated probabilities of four cases in Table 3.1 with an SNR of 30 dB at the input of the channel. The sum of four probabilities is equal to 1 and the ratio of dLev-UP to dLev-DN is 3:1. The graphs have an asymmetric characteristic around the lock point, and the reason can be explained by the eye diagrams below the probability graph. As the phase shifts from the lock point to the left, the pre-cursor ISI becomes smaller, so pattern 1 and 0 are gathered at one level, respectively. On the other hand, as the sampling phase moves from the lock point to the right, the pre-cursor ISI increases, and accordingly, pattern 1 and 0 of the eye diagram are clearly divided into two parts respectively. As a result, when the phase is shifted to the far right, the probabilities of *dLev*-UP, *dLev*-DN, and phase-UP are saturated to about 0.5, 0.5/3, and 1-0.5-0.5/3, respectively. The phase detector (PD) gain obtained by the difference between two probabilities phase-UP and phase-DN is shown in Fig. 3.6(b). When SNR decreases from 30 dB o 10 dB, the PD gain decreases and becomes more linear without saturation.



(a)



(b)

Fig. 3.5 (a) Samples for BB-CDR and MET-CDR. (b) Process of convergence.

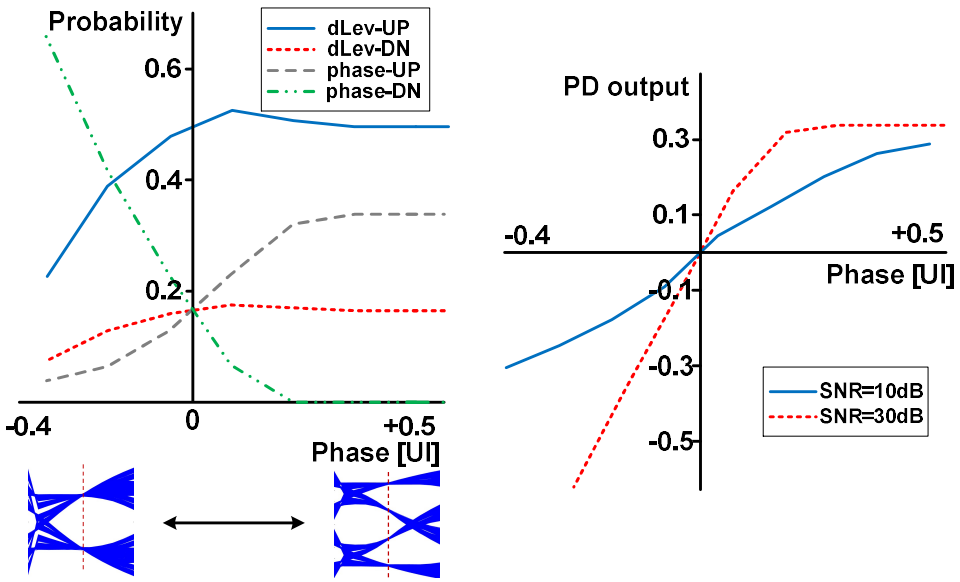
Table 3.1 Adaptation Algorithm

L-error	R-error	Operation
-	-	**dLev DN & update w_n**
-	+	**Phase DN**
+	-	**Phase UP**
+	+	**dLev UP & update w_n**

In determining ΔT between two samples, there is a trade-off. With smaller ΔT, we can find more accurate peak of eye height as shown in Fig. 3.7(a). Also, DFE coefficients become more accurate with smaller ΔT. To simplify the hardware implementation, one DFE is used for two samples and it converges on the average of the two post cursors. Fig. 3.7(b) shows the simulated difference between h_1 and w_1 according to ΔT. On the other hand, large ΔT is preferred to obtain a large PD gain and improve the jitter tracking capability [36] , [37] . With large ΔT, the probability that the signs of the L/R-error are opposite becomes large. The sampling phase could be updated more frequently and that results in a larger PD gain. Fig. 3.7(c) shows the simulated PD gain according to ΔT. From this trade-off, there exists an optimal ΔT that minimizes BER.

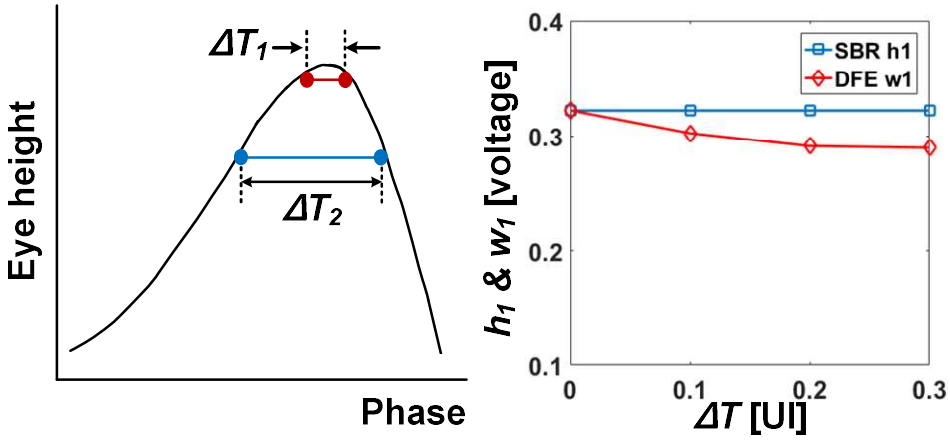
To precisely detect the peak and the slope of the eye height curve, the step sizes (μ) for phase and dLev should be small. For general optimum searching algorithms, two pathological cases should be considered as follows when small step sizes are used.

First, if the flat region of the search area is wider than the step size, the algorithm can be stuck at an edge or wander due to noise. If there is little or no ISI, the eye diagram becomes close to a rectangular shape with a zero slope. In this work, we assumed a band-limited analog front-end to avoid this zero slope condition [38] . Second, it can be trapped at the local optimum depending on the initial values. However, the search area or the eye height curve is a smooth convex function for typical SBRs, so the local optimum and the global optimum are the same.

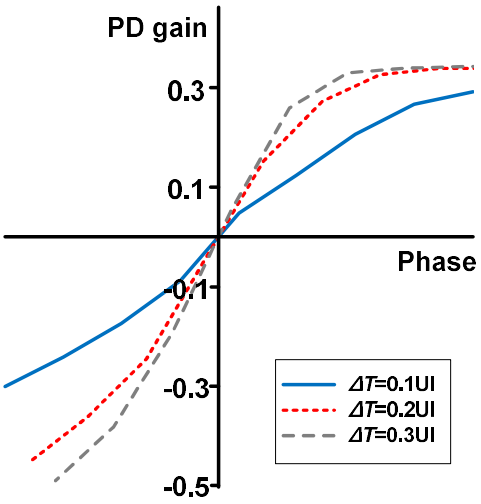


(a) (b)

Fig. 3.6 (a) Simulated probabilities of four cases in Table 3.1, and eye diagrams at far left and far right from lock point. (SNR=30 dB, ΔT=0.1UI, weighting factor=1:3) (b) Simulated PD gain according to SNR.



(a) (b)



(c)

Fig. 3.7 (a) Accuracy of peak finding. (b) Simulated difference between h_1 and w_1. (c) Simulated PD gain according to ΔT (SNR=10 dB, weighting factor=1:3).

3.2.3 Architecture and implementation

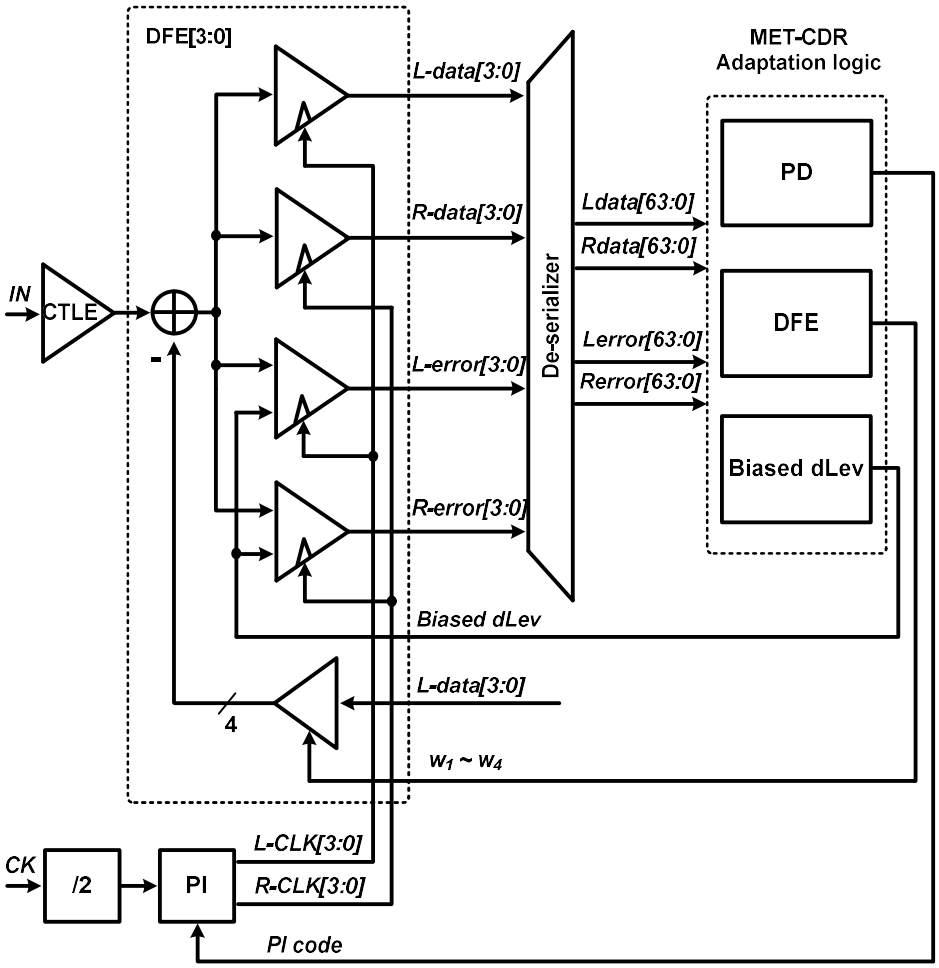


Fig. 3.8 Architecture of the proposed receiver with MET-CDR.

Fig. 3.8 shows the overall architecture of the proposed receiver with MET-CDR. It consists of a CTLE with moderate boosting gain, a four-tap quadrature DFE generating L/R-data and L/R-error, and an adaptation logic to update sampling phase, biased dLev and DFE coefficients. Four-phase clocks are generated from the forwarded differential clocks with a divider [39] and a phase interpolator (PI) [40] . This architecture is much simpler than the BER or EOM based architectures. It only adds one sampler compared to the BB-CDR. The BB-CDR requires three samplers (one data sampler, one edge sampler and one error sampler) and the proposed MET-CDR requires four samplers (two data samplers and two error samplers) per one clock phase.

For the DFE design, direct feedback of h_1 is adopted to reduce the area and power consumption [41] , [42] . To relax the delay constraint of the first tap, the output of the strong arm latch in the return-to-zero (RZ) format is directly used [43] . The precharge state of the RZ signal is behind the sampling time for the first tap since the DFE is based on a quarter-rate clocking. For the second to the fourth taps whose delay constraints are relaxed, non-return-to-zero (NRZ) signals after RS latches are used.

The implementation of the adaptation algorithm is shown in the column ‘with L/R-data samplers’ in Table 3.2. It is the detailed version of the algorithm shown in eye, or the zero crossing of the DFE output, is located between two sampling phases as shown in Fig. 3.9. Even if the errors for dLev are calculated assuming L/R-data are the same in ‘case3’, it does not cause a significant error because two data are almost zero. Also, if the state changes from ‘case3’ to ‘case4’ as dLev converges,

Table 3.2 Adaptation Algorithm for One DFE for Two Samples

Case	L/R-data	L/R-error	Operation	
			With L/R-data samplers	With L-data sampler only
1	Same	Same	Update DFE (dLev, w_n)	Update DFE (dLev, w_n)
2	Same	Different	Update PD	Update PD
3	Different	Same	Ignore	Update DFE (dLev, w_n)
4	Different	Different	Update PD	Update PD

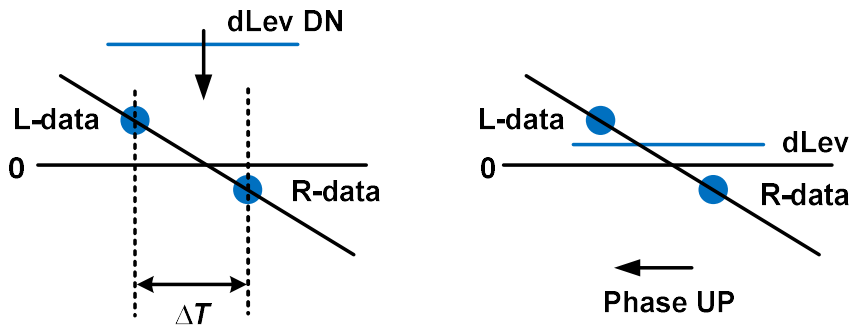


Fig. 3.9 Illustration of (a) case3 and (b) case4 in Table 3.2.

the sampling position moves inside of the eye by updating PD. After leaving the eye edge, the algorithm runs only in ‘case1’ and ‘case2’ until the adaptation terminates. The operation with three samplers (one data sampler, two error samplers) was verified with simulation, and it indicates that the proposed MET-CDR can be implemented with the same hardware as the BB-CDR.

To simplify the hardware implementation, L-data is used as a recovered data output. It means the output sample is $\Delta T/2$ away from the best sampling point, and this trade-off is already described in Chapter3.2.2. For more precise operation for large ΔT, a center phase for the data sampler can be used by selecting the center code between L/R codes in PI.

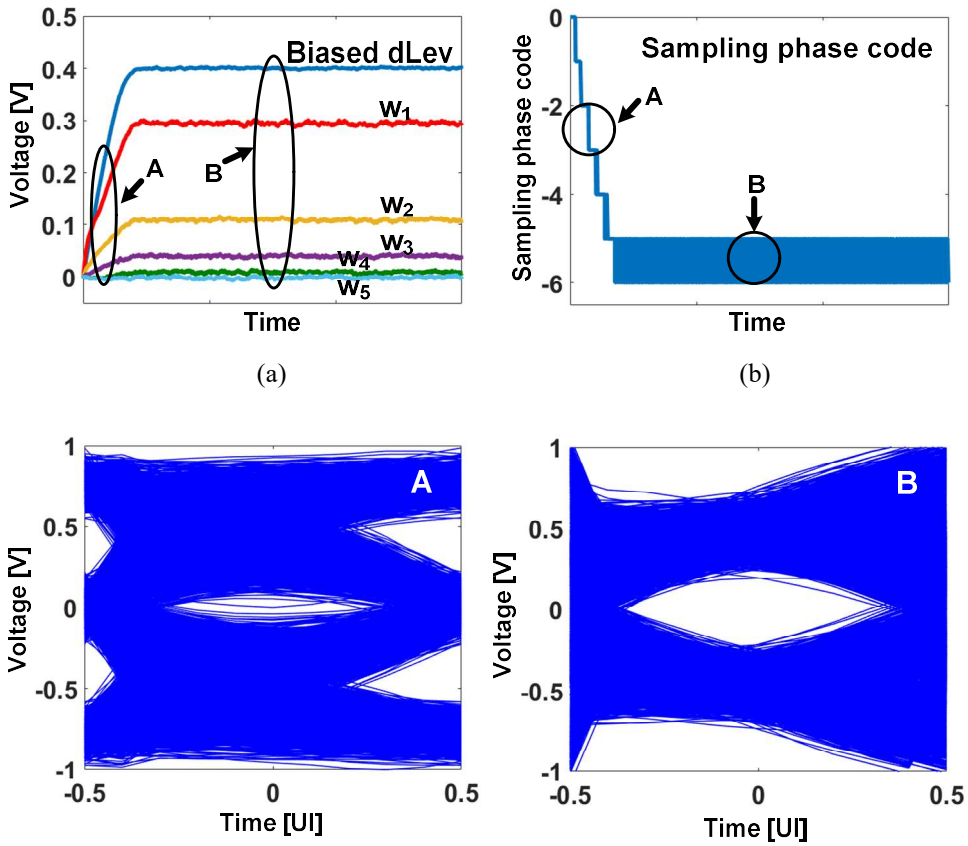
3.2.4 Verification of the Algorithm

Operation of the proposed MET-CDR is verified by MATLAB simulation. To check the convergence behavior for a noisy environment, additive white Gaussian noise (AWGN) channel having an SNR of 10 dB is used. Fig. 3.10(a) and (b) show the process of convergence of the DFE coefficients, the biased dLev and the sampling phase code, respectively. There are 10 sampling phase codes per UI in the simulation environment and the negative values of the sampling codes mean that the locked phase has moved left from the initial phase. A and B represent certain states during and after convergence. Simulated eye diagrams with 5000 data samples are shown in Fig. 3.10(c) and (d). The eye height is improved in the B state because of the canceled post-cursor ISI from DFE adaptation and the reduced pre-cursor ISI from searching the optimal phase.

The converged results of the MET-CDR are compared with that of the BB-CDR by applying five channels for simulation. Each SBR and the eye height of the channels are shown in Fig. 3.11(a) and (b). To precisely compare the simulated results and the predicted results from each SBR, AWGN is not included. The circles and squares in Fig. 3.11(b) represent the locked phases of the BB-CDR and the MET-CDR for each channel. With MET-CDR, the locked phase appears at the maximum eye height in all the channels. The resulting eye heights are summarized in Fig. 3.11(c). As the loss of the channel and pre-cursor ISI increase, the amount of improvement by MET-CDR also increases: 10% for ‘channel1’ and 53% for ‘channel5’. Simulated eye diagrams with ‘channel3’ are shown in Fig. 3.11(d). In the proposed MET-CDR,

the eye height is increased by searching the maximum eye height. In addition, the residual ISI is reduced and the reason for improved DFE adaptation will be described in Chapter3.2.5.

As mentioned earlier, we assumed an ideal DFE with large enough number of taps. Fig. 3.12(a) shows the simulated eye heights and locked phases with insufficient number of DFE taps with 'channel3'. Unlike the BB-CDR, the locked phases of the MET-CDR could be affected by the number of taps because it operates with signal after DFE. The overall eye height is reduced at each locked phase when the number of taps is reduced. Especially, the amount of improvement in eye height by the MET-CDR is also reduced as summarized in Fig. 3.12(b). However, even with the imperfect DFE, the MET-CDR is always better in eye height than the BB-CDR. For typical SBRs, the slope of the pre-cursor ISI is sharper than the slope of the post-cursor ISI. So, when switching from the locked phase of the BB-CDR to the locked phase of the MET-CDR, the eye height is improved because the increase in post-cursor ISI is smaller than the decrease in pre-cursor ISI.



(a) (b)

(c) (d)

Fig. 3.10 Simulated process of convergence. (a) DFE coefficients and biased dLev.

(b) Sampling phase code. Eye diagrams (c) during and (d) after adaptation.

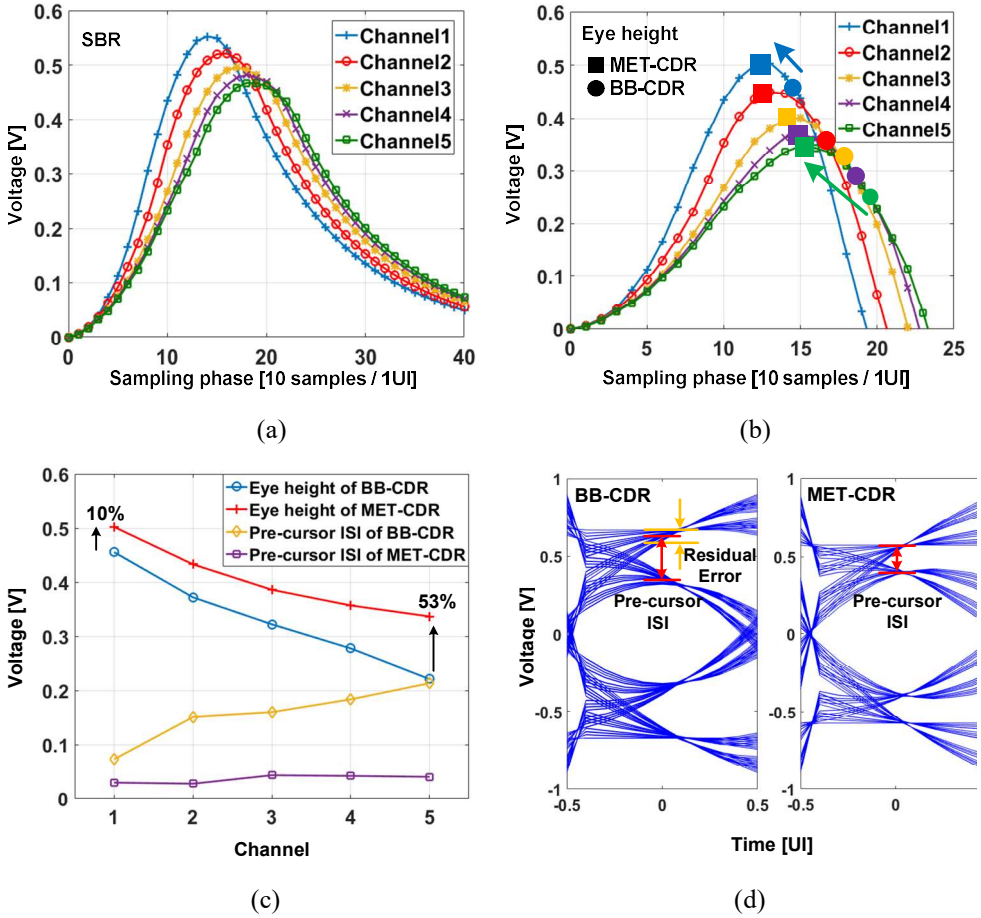
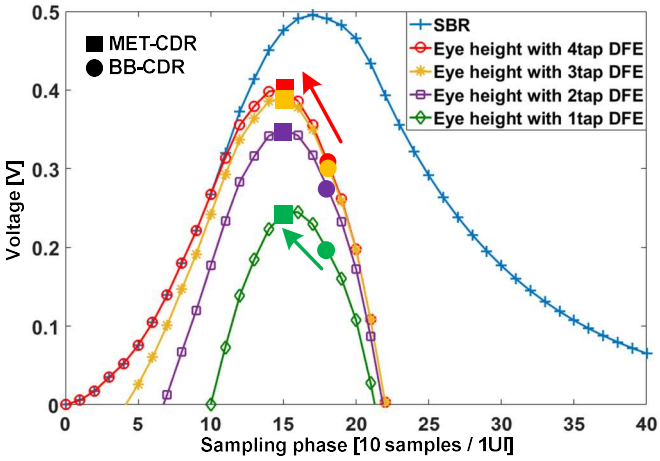
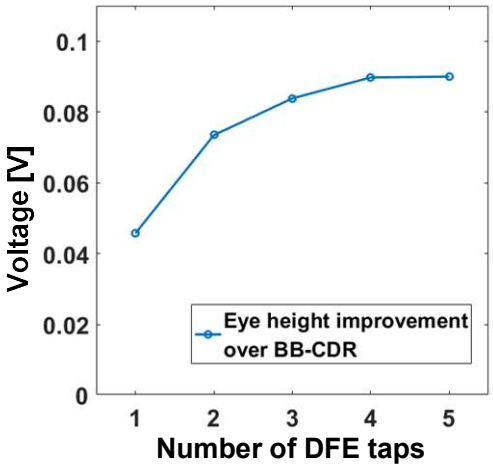


Fig. 3.11 Simulated converged results of the BB-CDR and the proposed MET-CDR. (a) SBRs and (b) eye height curves of five channels. (c) Resulting eye height and pre-cursor ISI after adaptation. (d) Eye diagrams with ‘channel3’.



(a)



(b)

Fig. 3.12 (a) Simulated eye heights with insufficient number of DFE taps. (b) The amount of improvement by MET-CDR over BB-CDR according to the number of DFE taps.

3.2.5 Analysis on the Biased Data-Level

3.2.5.1 Improving Accuracy of the SS-LMS with Biased dLev

By using the proposed biased dLev, there is another advantage as well as the extraction of eye height information described above. The accuracy of the SS-LMS algorithm can be improved with biased dLev [44] . The SS-LMS algorithm is widely used for the DFE adaptation because it is easy to implement [21] , [45] . The conventional SS-LMS adaptation equation for dLev and the n-tap coefficient are expressed as follows [21] :

$$\boldsymbol{dLev[k+1] = dLev[k] + \mu_{dLev} \times sign(err[k] \times d[k])} \quad (3.15)$$

and

$$\boldsymbol{w_n[k+1] = w_n[k] + \mu_{w_n} \times sign(err[k] \times d[k-n])} \ . \quad (3.16)$$

Instead of the absolute value of error, it only detects the polarity of the error. When there is no pre-cursor ISI, SS-LMS achieves the same result as LMS because of the zero-forcing effect. However, when there is pre-cursor ISI and dLev is fixed to h_0, the coefficients of the DFE may not perfectly cancel the post-cursor ISI [44] . In this work, further analysis on wandering of not only the DFE coefficients but also dLev

is conducted considering AWGN.

Fig. 3.13 (a), (b), (c) and (d) show four cases of eye diagrams for channels with one pre-cursor ISI with the typical 1:1-dLev. Four lines A, B, C and D indicate the pattern boundaries of the eye diagram. These four cases are detected as the same states in the SS-LMS algorithm because the error polarity, or the second terms of (3.15) and (3.16), are the same. So, the DFE adaptation could be finished with dLev error and residual DFE error. The possible amount of error after DFE adaptation can be expressed as follows:

$$ERR_{dLev} + ERR_{w_n} \leq |h_{-1}| \ , \tag{3.17}$$

where

$$ERR_{dLev} = |h_0 - dLev| \tag{3.18}$$

and

$$ERR_{w_n} = \sum_{n=1}^{\infty} |h_n - w_n|. \tag{3.19}$$

Fig. 3.13(a) shows the ideal position of 1:1-dLev and ideal DFE adaptation. In the absence of DFE error, or ERR_{wn}, dLev can be located anywhere between the minimum and the maximum values as shown in Fig. 3.13(b) and (c). In these cases, ERR_{dLev} can be maximized. When dLev is located between the minimum and the maximum values as shown in Fig. 3.13(d), ERR_{wn} can have any value within the

area marked with diagonal patterns. As a result, the eye height can be reduced due to the residual ISI error. In other words, dLev and w_n can not only converge on different values but also wander within the region satisfying (3.17) according to many environments such as the initial values, loop bandwidth, high frequency noise and change of temperature. With 1:3-dLev shown in Fig. 3.13(e), ERR_{wn} is minimized because ERR_{dLev} is intentionally maximized. In this case, the dLev is fixed and each w_n is converge on the fixed optimal value. The 3:1-dLev also guarantees optimal DFE adaptation [44] , but 1:3 is chosen in this work to extract the effective eye height information. The second terms of (3.15) and (3.16) and the resulting DFE adaptation behaviors for eight cases with three consecutive data are summarized in Table 3.3. For simplicity, only the first post-cursor is considered in this table. With the typical SS-LMS with 1:1-dLev, the sign equilibrium is satisfied even if w_n is not optimal. With 1:3-dLev, the sign equilibrium is satisfied only when w_n is at its optimum value. A description about the column named '1:7-dLev' is given in the next section.

In a real environment with AWGN, the inaccuracy of the typical SS-LMS is reduced because the possible wandering region of dLev and w_n in (3.17) is reduced by the amount of noise. However, for channels with large pre-cursor ISI, the eye opening and BER can be further improved by adopting the biased dLev.

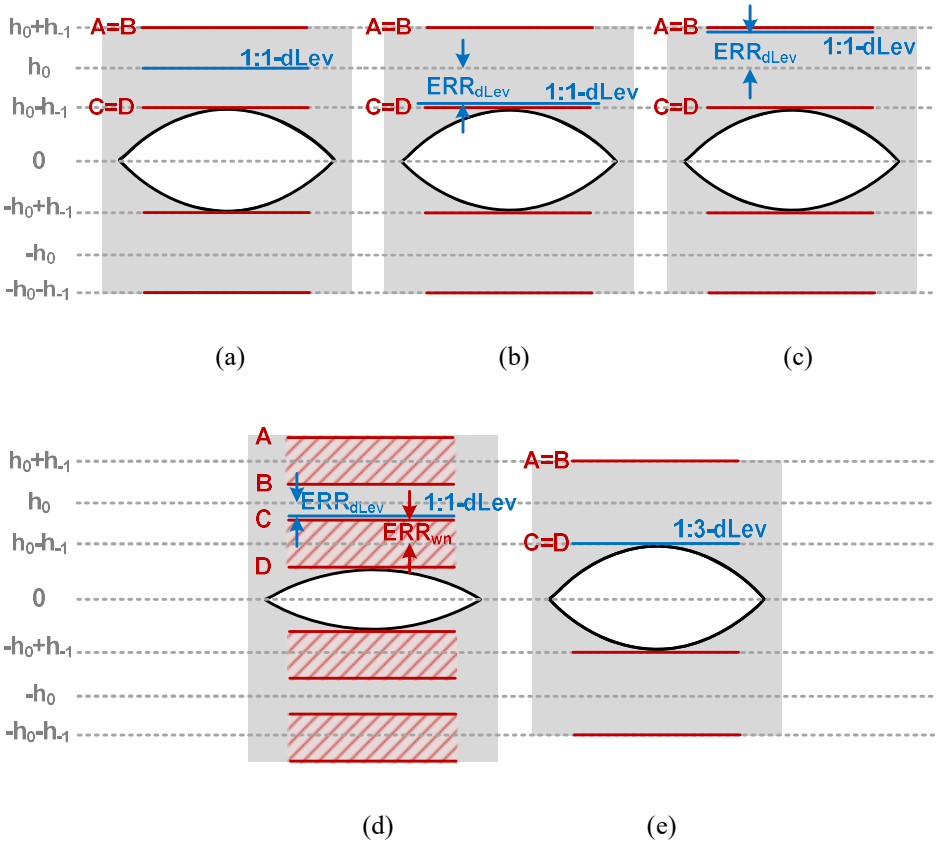


Fig. 3.13 (a) Ideal position of 1:1-dLev and ideal DFE adaptation. (b) Minimum possible and (c) maximum possible position of 1:1-dLev in the absence of DFE error. (d) Resulting error in both 1:1-dLev and DFE adaptation. (e) 1:3-dLev is fixed and correct DFE adaptation is obtained.

Table 3.3 Operation of the SS-LMS for Three Consecutive Data for Channels with One Pre-cursor ISI

Status of DFE coefficient	**Posi-tion**	***d[k-1]***	***d[k]***	***d[k+1]***	***Sign(err[k] x d[k])***			***Sign(err[k] x d[k-1])***			***w_1***		
					1:1-dLev	**1:3-dLev**	**1:7-dLev**	**1:1-dLev**	**1:3-dLev**	**1:7-dLev**	**1:1-dLev**	**1:3-dLev**	**1:7-dLev**
w_1 < optimal	**A**	+	+	+	+	+	+	+	+	+	**Keep current value**	**UP**	**UP**
	B	-	+	+	+	+	+	-	-	-			
	C	+	+	-	-	+	+	-	+	+			
	D	-	+	-	-	-	-	+	+	+			
w_1 = optimal	**A=B**	+/-	+	+	+	+	+	+/-	+/-	+/-	**Keep current value**	**Keep current value**	**Keep current value**
	C=D	+/-	+	-	-	+/-	+/-	+/-	+/-	+/-			
w_1 > optimal	**A**	-	+	+	+	+	+	-	-	-	**Keep current value**	**DN**	**DN**
	B	+	+	+	+	+	+	+	+	+			
	C	-	+	-	-	+	+	+	-	-			
	D	+	+	-	-	-	-	-	-	-			

3.2.5.2 Effect of the Weighting Factor

The right weighting factor for the biased dLev is determined by the number of the pre-cursors as described earlier. In this chapter, the effect of the weighting factor on adaptation of DFE and MET-CDR is discussed.

First, assume that the number of pre-cursors is mispredicted (less than the actual number) and a smaller weight (α) is used. In this case, as described in the previous section, it results in inaccurate dLev and DFE coefficients because of the wandering. Even if AWGN reduces the DFE wandering, the MET-CDR converges on the non-optimal phase due to the wrong pre-cursor information, thus increasing the BER.

Second, assume that the number of pre-cursors is overestimated. The columns named '1:7-dLev' in Table 3.3 show the case when 1:7-dLev is used for channels with only one pre-cursor ISI. Although 1:7-dLev is lower than the optimal 1:3-dLev, the pattern boundary D is still below 1:7-dLev before DFE adaptation. So, the error signs for pattern boundaries A, B, C and D are the same as that of 1:3-dLev case and the sign equilibrium is satisfied only with the optimal w_n. Also, there is little impact on searching the maximum eye height because the vertical difference between 1:3-dLev and 1:7-dLev caused by AWGN is almost constant regardless of the sampling phase and the peak positions appear at the same phase. However, unconditionally using large α may degrade the jitter tracking capability. Since the error signs are determined by the position of dLev, the lowered dLev with larger α than its optimal value results in the lowered probability of 'case2' and 'case4' in Table 3.2, leading to a reduced PD gain. The simulated results for a channel with an optimal weighting

factor of 1:3 are shown in Fig. 3.14. The results of the above analysis are summarized in Table 3.4.

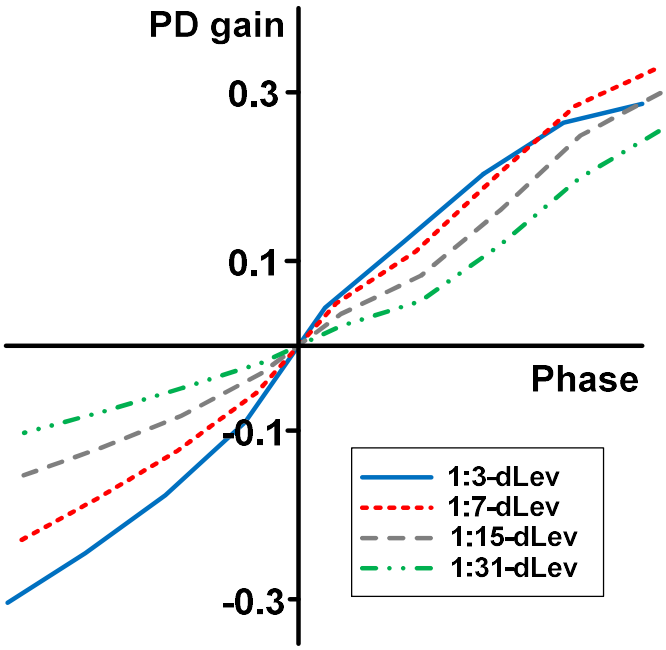


Fig. 3.14 PD outputs according to weighting factors $1:2^{N+1}-1$ (N=1, 2, 3, 4) for a channel whose optimal weighting factor is 1:3. (SNR=10 dB, ΔT=0.1UI).

Table 3.4 Operation of the DFE and the MET-CDR According to the Weighting Factors

Number of Pre-cursor ISI	Weighting factor of dLev		DFE (dLev, w_n)	MET-CDR	
	Desired	Setting		Searching max. eye height	Jitter tracking capability
0	1:1	1:1	Optimal	Optimal	
		1:3	Optimal	Optimal (peak of 1:3-dLev = peak of 1:1-dLev)	Reduced
		1:7	Optimal	Optimal (peak of 1:7-dLev = peak of 1:1-dLev)	Reduced
1	1:3	1:1	Wander within $ERR_{wn} + ERR_{dLev} \leq \lvert h_{-1} \rvert$	Non-optimal	
		1:3	Optimal	Optimal	
		1:7	Optimal	Optimal (peak of 1:7-dLev = peak of 1:3-dLev)	Reduced
2	1:7	1:1	Wander within $ERR_{wn} + ERR_{dLev} \leq \lvert h_{-1} \rvert - \lvert h_{-2} \rvert$	Non-optimal	
		1:3	Wander within $ERR_{wn} + ERR_{dLev} \leq \lvert h_{-2} \rvert$	Non-optimal	
		1:7	Optimal	Optimal	

3.3 Expansion of MET-CDR to PAM4 signaling

In the previous chapters, MET-CDR was described for NRZ (non-return-to-zero) and chip was also implemented for NRZ signal. In this chapter, the expansion of the proposed MET-CDR to PAM4 (pulse amplitude modulation) signaling is described. The operations are verified by simulation.

3.3.1 MET-CDR with PAM4

PAM4 signaling can transmit 2 data at a time by dividing a symbol into 4 levels [46] -[48] . Nyquist frequency is 1/2 compared to NRZ and eye height is 1/3, so it is effective to use PAM4 than NRZ when there is a channel loss gain of about 9.5dB at a frequency reduced to 1/2.

As mentioned in Chapter3.2.1, the expressions of the actual eye heights of NRZ and PAM4 calculated using post-cursor ISI and pre-cursor ISI are as follows:

$$\boldsymbol{EyeHeight_{NRZ} = h_0 - \sum_{m=1}^{\infty}|h_m| - \sum_{n=-\infty}^{-1}|h_n|} \quad (3.20)$$

and

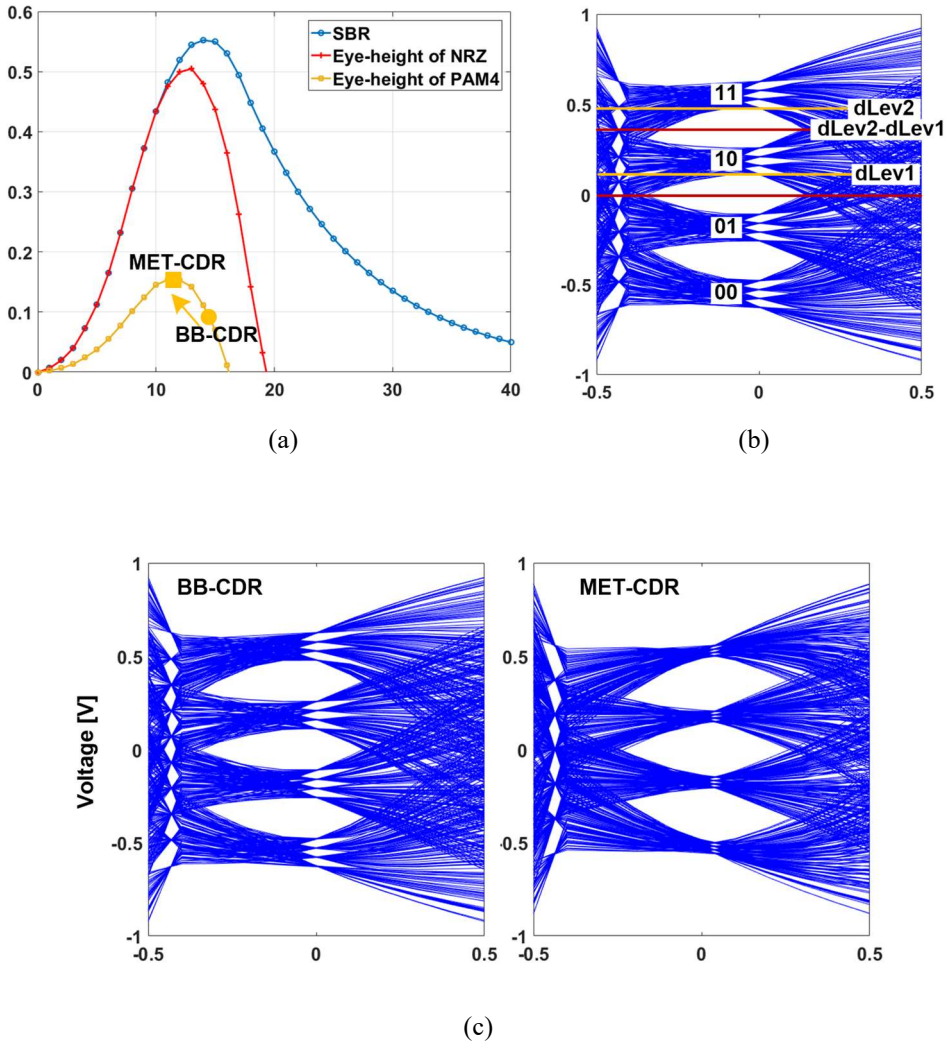
$$\boldsymbol{EyeHeight_{PAM4}} = \frac{\boldsymbol{h_0}}{\boldsymbol{3}} - \sum_{\boldsymbol{m=1}}^{\infty}|\boldsymbol{h_m}| - \sum_{\boldsymbol{n=-\infty}}^{\boldsymbol{-1}}|\boldsymbol{h_n}|. \quad (3.21)$$

So, in PAM4, ISI has the same effect as in the value in SBR, but the main cursor only affects as much as 1/3. It means that the eye height of PAM4 signal is relatively more affected by pre-cursor ISI than NRZ. The SBR and eye height curves for NRZ and PAM4 are shown in Fig. 3.15(a).

For MET-CDR implementation, basically, the same algorithm as NRZ is used. There are two changes. The eye height curve function is changed from (3.20) to (3.21). And 4 cases are generated unlike NRZ where 2 cases are generated for one pre-cursor, so the weighting factor is changed from 1:3 to 1:7 for 1 pre-cursor ISI.

As shown in Fig. 3.15(b), the bottom part of data 11 is defined as dLev2 and the bottom part of data10 is defined as dLev1. The algorithm is designed to maximize dLev1. When data is 10 and 11, dLev1 and dLev2 are updated respectively. When the level separation mismatch ratio (RLM) [48] is assumed to be 1, data 10 and 11 are determined with dLev2-dLev1.

The eye diagrams after phase search with BB-CDR and MET-CDR for PAM4 is shown in Fig. 3.15(c). It is confirmed that the pre-cursor ISI is reduced and the eye height is increased with MET-CDR because it is locked at the position pulled forward than the locked position in the BB-CDR.



(a) (b)

(c)

Fig. 3.15 (a) SBR and eye height curves for NRZ and PAM4. (b) Definition of dLev1 and dLev2 for PAM4. (c) Simulated eye diagrams for BB-CDR and MET-CDR after CDR lock

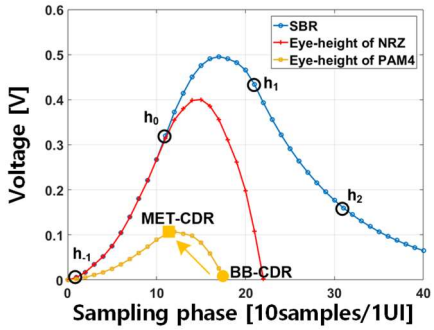
3.3.2 Considerations for PAM4

This chapter describes points to consider when applying MET-CDR to PAM4. As seen earlier, PAM4 is more sensitive to pre-cursor ISI than NRZ, and the position locked by MET-CDR is more likely to be pulled forward than NRZ.

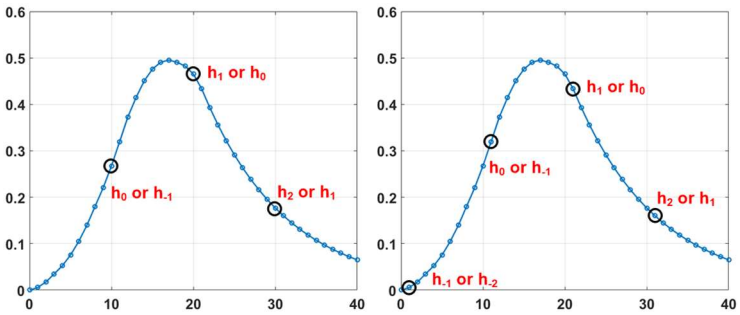
As an example, let's consider a channel that is lossy enough to close the PAM4 eye with BB-CDR. As shown in Fig. 3.16(a), the phase where h_0 is pulled forward to be larger than h_1 is the theoretical optimal phase. In this situation, confusion may occur in DFE operation.

For a given channel SBR in Fig. 3.16(a), the results of observing the operation of the DFE as sweeping the phase with CDR off are shown in Fig 3.16(b) and (c). There are 10 samples in 1UI. In the case of Figure 3.16(b), the same point in SBR can converge to h_0 or h_1 depending on the environment. In other words, each point converging to $h_0/h_1/h_2$ can be converged to $h_{-1}/h_0/h_1$, that are the points pushed behind one UI. When observing h_0 backward by one sample, all of the results in Fig. 3.16(c) are recognized as h_0 at the intended phase.

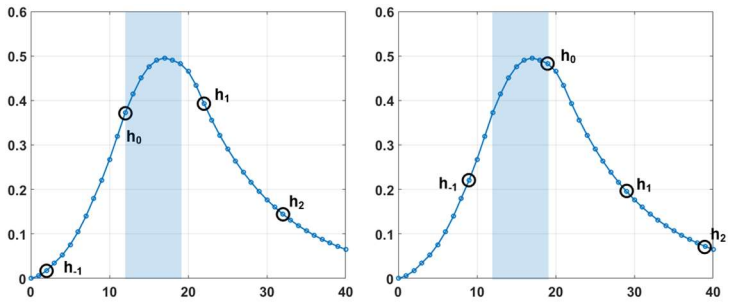
It can be roughly expressed that DFE operates stably in the region that satisfying $h_0>h_1$, but since DFE is a nonlinear block, the results can be changed by initial value, noise, bandwidth of coefficients, and so on. Therefore, increasing the ratio of main cursor to pre-cursor with the help of FFE in transmitter and CTLE is helpful for stable operation of MET-CDR and DFE.



(a)



(b)



(c)

Fig. 3.16 (a) Lossy channel and eye height curves for NRZ and PAM4. (b) h_0 positions where confusion can occur in DFE operation. (c) Stable DFE operation cases.

Chapter 4

Measurement Results

The proposed MET-CDR has been designed and fabricated in a 28 nm CMOS process as shown in Fig. 4.1. The silicon area of the CDR core including CTLE, DFE, deserializer (DES), clock divider, PI and digital block is 0.089mm^2. The receiver operates up to 26 Gb/s with a PRBS7 pattern. The measured power consumption of the CDR core is 87 mW (CTLE /DFE/DES/part of clock repeater: 62 mW, clock buffer/clock divider/PI/part of clock repeater: 25 mW).

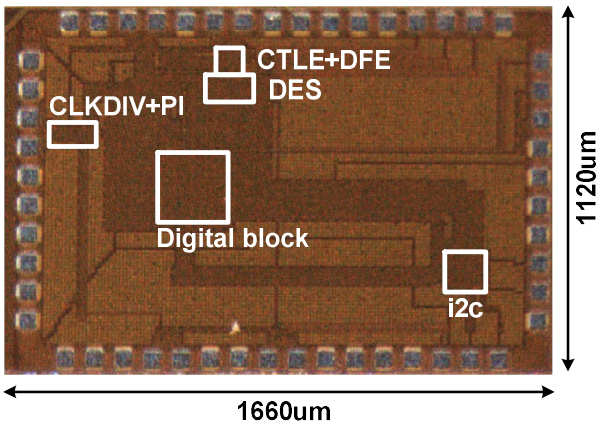
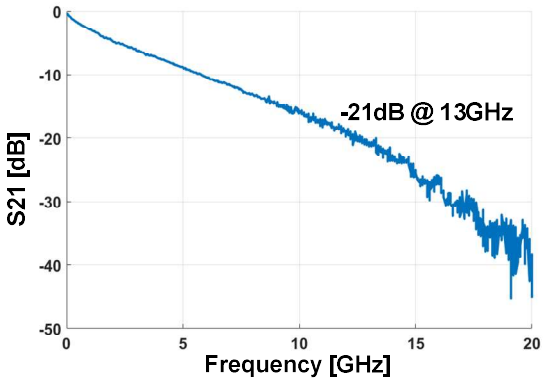
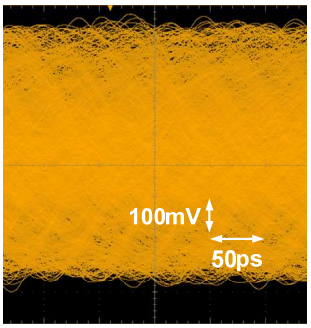


Fig. 4.1 Die photograph.

Fig. 4.2 (a) shows the measured S21 of the channel. The loss at the Nyquist frequency of 13 GHz is 21 dB. Measured eye diagram at the end of the channel is shown in Fig. 4.2(b). Including 2.5 dB of additional PCB trace loss obtained from the HFSS simulation, the total loss is about 23.5 dB.



(a)



(b)

Fig. 4.2 (a) Measured channel frequency response. (b) Measured eye diagram at the end of the channel.

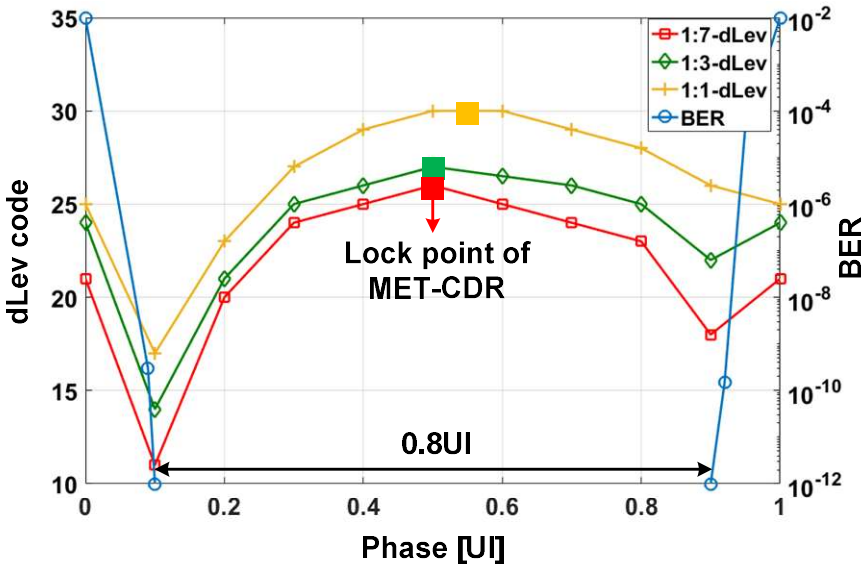
Fig. 4.3 shows three dLevs with different weighting factors (1:1, 1:3, 1:7) and the bathtub curve with 1:7, by sweeping sampling phase with CDR loop off. The *dLev*s are measured with 5 bit digital code (0~31). After sweeping sampling phase, *dLev* code is measured once again with CDR loop on to check the lock positions indicated by squares in Fig. 4.3. MET-CDR is locked at the maximum point of *dLev*. There are three important points we can analyze from these results.

First of all, the peak positions in x-axis of dLevs are affected by pre-cursor ISI. In the results with 10 Gb/s as shown in Fig. 4.3(a), the peak of dLev with the ratio of 1:3 appears earlier than the peak of dLev with the ratio of 1:1, because of the first pre-cursor ISI, h_{-1}. However, the peak of 1:7-dLev is the same as the peak of 1:3-dLev, because there is no second pre-cursor ISI, h_{-2}. On the other hand, in Fig. 4.3(b) with 26 Gb/s, three peaks appear sequentially because there exist two pre-cursors, h_{-1} and h_{-2}. In addition, if there is no pre-cursor ISI, the three peaks will appear at the same position in x-axis. The second point is that the vertical differences between the three levels are affected by not only the pre-cursor ISI, but also noise such as power supply noise, white Gaussian noise, reflection, and so on. For two dLevs with the ratio of 1:3 and 1:7 in Fig. 4.3(a), the vertical difference appears because of the noise even if there is no h_{-2}. The last important point is that the locked phase of the MET-CDR, or the peak phase of 1:7-dLev, matches well with the center of the bath-tub curve where BER is less than 10^{-12}. It proves that the proposed MET-CDR effectively finds the optimal sampling phase to minimize BER. It is noteworthy that the peak of 1:1-dLev deviates from the center of the bathtub curve in Fig. 4.3(b). Therefore, the locked phase of the BB-CDR, which generally appears later than the peak of 1:1-dLev, also deviates from the center of the bathtub curve.

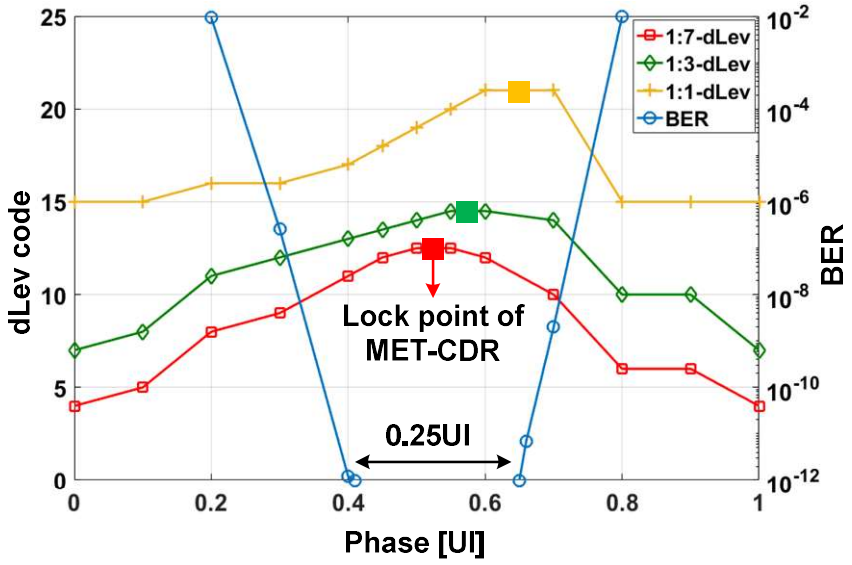
Fig. 4.4 shows two bathtub curves with different weighting factors in 26 Gb/s. Since the results are obtained with CDR loop off, it does not contain the effect of MET-CDR, and it only shows the accuracy of the DFE coefficients. The ratio of 1:7 is the optimal setting because there are two pre-cursors as shown in Fig. 4.3(b). With biased dLev, BER is reduced due to increased accuracy of w_n and dLev over conventional SS-LMS. The resulting eye-opening is increase by 2% of a UI with biased dLev.

Fig. 4.5(a) shows the simulated jitter tolerance in 26 Gb/s. In the forwarded clocking architecture, the jitter tolerance corner frequency that exhibits slope of -20 dB/dec is obtained by the skew between two channels for data and clock. With the measured T_{skew} of 1.87 ns and the analysis of jitter transfer characteristics [28] , [31] , we can obtain the simulated jitter tolerance and the corner frequency is about 100 MHz. If T_{skew} is reduced by matched channels or delay locked loop (DLL), the jitter tolerance can be improved overall as the corner frequency is shifted.

Fig. 4.5(b) shows the measured jitter tolerance in 26 Gb/s within the equipment limit represented with box in Fig. 4.5(a). The measured result with the weighting factor of 1:7 matches well with the simulation result. When we change the weighting factor from 1:7 to 1:3, jitter tolerance is lowered because the second pre-cursor ISI, h_{-2}, is not taken into account in the weighting factor of 1:3.



(a)



(b)

Fig. 4.3 (a) Measured three dLevs and bathtub curves for (a) 10 Gb/s and (b) 26 Gb/s.

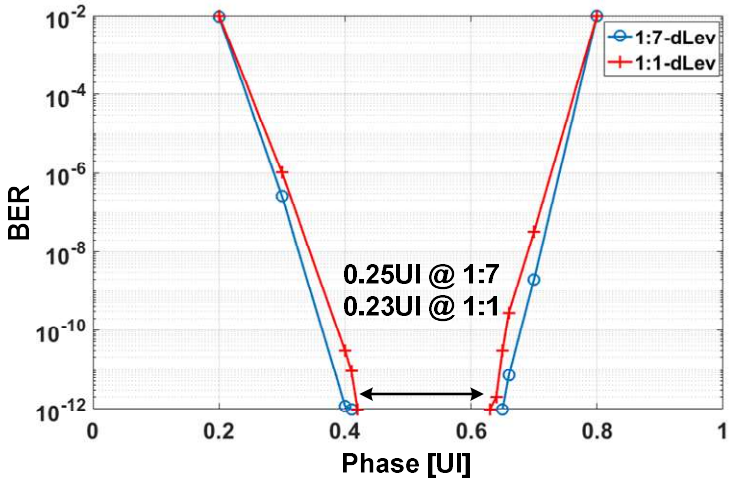
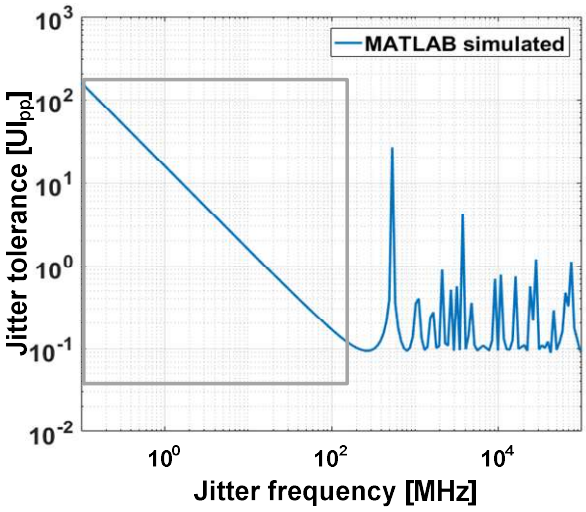
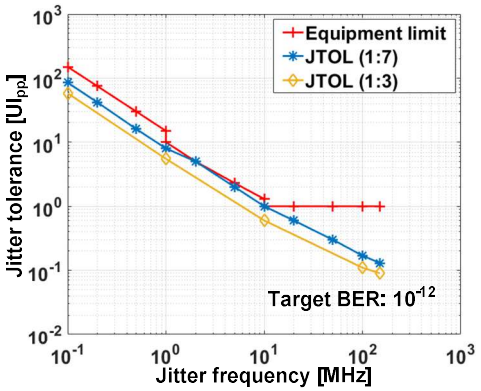


Fig. 4.4 . Measured bathtub curves with conventional SS-LMS (1:1-*dLev*) and proposed biased *dLev* (1:7-dLev) in 26 Gb/s.

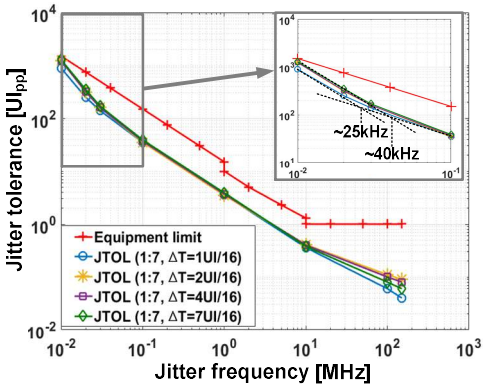
Fig. 4.5(c) shows the effect of ΔT. To see the slope of -40 dB/dec below the CDR loop bandwidth within the equipment limit, we intentionally degraded the performance of the receiver by varying the bias current. As explained earlier, there exists a trade-off in determining ΔT. With ΔT of 2UI/16, the loop bandwidth is increased from 25 kHz to 40 kHz and the improvement is also visible in the high frequency region.



(a)



(b)



(c)

Fig. 4.5 (a) Simulated JTOL. (b) Measured JTOL according to weighting factors. (c) Measured JTOL according to ΔT.

Fig. 4.6 shows the 13 GHz input clock and the recovered 6.5 GHz clock. RMS jitter of the recovered clock is about 1.85 ps. Table 4.1 shows comparisons between various types of CDR architectures that search for optimal sampling phase.

Table 4.1 shows the comparison between various types of CDR architectures that search for optimal sampling phase. For MET-CDR, the added hardware for timing adaptation compared to BB-CDR is only one sampler per clock phase. It is the simplest hardware implementation among four architectures in this table, and even this one sampler can be removed. Processing time is in order of μs, which is an improvement over previous works that require BER counting or eye sweeping time. One-time adaptation is possible without iteration, and all operations are implemented on-chip without off-chip assistance. Few examples of CDRs finding the optimal sampling phase are implemented with the same process. Therefore, all of the works in Table 4.1 used different technologies, and it is rather difficult to fairly compare operating speed, channel loss, and power consumption.

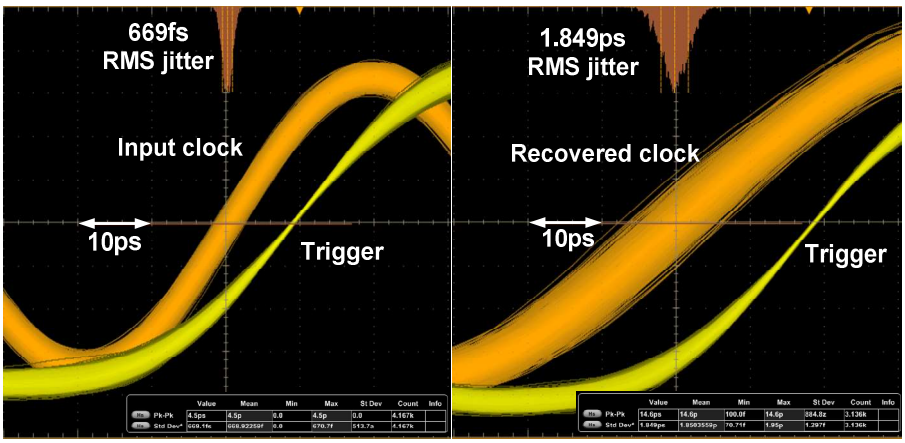


Fig. 4.6 Measured input clock and recovered clock.

Table 4.1 Comparison of Optimal Timing Adaptation CDRs

	[4]	[5]	[7]	This work
Method	BER (basic hill-climbing)	BER (stochastic hill-climbing)	EOM (stochastic sigma-tracking)	**Max-eye tracking**
Added hardwares for timing adaptation compared to BB-CDR	Sampler, XOR gate, BER counter, Logic for iteration	Sampler, XOR gate, BER counter, Logic for iteration	Sampler, Probability counter, Logic for iteration	**One sampler** *
Processing time	'700x slower than SS-LMS' ** @ BER 10^{-4}	1 h 25 min *** @ BER 10^{-5} (2.55×10^{13} data cycles)	364 ms ** (1.02×10^{10} data cycles)	**< 20 us** *** **@ BER 10^{-12} (5.2×10^{5} data cycles)**
One-time adaptation	No (Iterative BER check)	No (Iterative BER check)	No (Iterative window check)	**Yes**
On-chip / off-chip clock recovery	Off-chip	Off-chip	N/A	**On-chip**
Process	90 nm	65 nm	40 nm	**28 nm**
Data-rate	6.25 Gb/s	5 Gb/s	28 Gb/s	**26 Gb/s**
Channel loss	14.28 dB	15 dB	25 dB	**23.5 dB** ****

* Although not implemented on this chip, this one sampler can also be removed as described in Chapter3.2.3.

** For equalizer adaptation time only. CDR adaptation time is not included. *** For both equalizer and CDR adaptation.

****Measured loss of channel: 21 dB, HFSS-simulated loss of additive PCB trace: 2.5 dB

Chapter 5

Conclusion

A maximum-eye-tracking CDR that finds the near-optimal sampling phase is proposed. Two samples detect the current eye height and the slope of the eye and thereby search for the phase where the eye height is maximized. The effective eye height is obtained by simply changing the weighting factor of UP and DN from error samplers in the presence of pre-cursor ISI. The typical weight factor is either 1:3 or 1:7 depending on the amount of pre-cursor ISI. This technique obtains the lock at the optimal phase without the need for complex hardware, long processing time, or assistance from external processors typically used in previous works based on BER or EOM.

Appendix A

MATLAB Code for Simulating Receiver with MET-CDR

In this section, a MATLAB code for simulating receiver with MET-CDR is presented. Before design the circuits with real transistors with H-spice and verilog, we can simply model the whole architecture and check the operation with MATLAB.

```
clear all; close all; clc;

%input
period=2^7-1; nu=1; repeat=2000; swing=1;
samples=10; halfUI=samples/2;
datalength=period*repeat; wholelength=datalength*samples;

in_prbs = idinput([period nu repeat],'prbs', [0 1], [-swing swing]);

for i=1:wholelength
   in(i)= in_prbs(ceil(i/samples)); %channel input
end

channelin=awgn(in, 15); %noise

%channelSBR
channelSBR = [...]; %single bit response for 10 times faster sampling
time. (too long to show )

sum=0; %for normalize channelSBR
for k=1:length(channelSBR)
   sum=sum+channelSBR(k);
end
channelSBR = channelSBR / sum;

rxin=filter(channelSBR,1,channelin); %signal after channel


%%initialize variables
slin=zeros(1,wholelength); %slicer(sampler) input
slout=zeros(1,datalength); %slicer(sampler) output
w1=zeros(1,datalength); w2=zeros(1,datalength); w3=zeros(1,datalength);
w4=zeros(1,datalength); w5=zeros(1,datalength); %DFE coefficients
level=zeros(1,datalength); %biased dLev
lerr=zeros(1,datalength); rerr=zeros(1,datalength); %left/right error
centertemp=zeros(1,datalength); %sampling phase (before quantization)
center=zeros(1,datalength);  %sampling phase (after quantization)

n=5; % n-tap DFE
intdelay=5; %internal delay from slout to DFE summer
levelr=3; %weighting factor
steplevel=0.0001; step1=0.0001; stepCDR=0.01; %step size
resol=1; %phase resolution (time spacing between L/R)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%feedback loop
for j=n+1:datalength-1

    k=center(j)+samples*(j-1);
    for m=(-1*samples)+1+intdelay:intdelay
        slin(k+m)=rxin(k+m)-w1(j)*slout(j-1)-w2(j)*slout(j-2)-
w3(j)*slout(j-3)-w4(j)*slout(j-4)-w5(j)*slout(j-5); %DFE summer operation
    end


    %decide data and error
    if(slin(k)>0 && slin(k+resol)>0)

        slout(j)=1;
        lerr(j)=slin(k)-level(j);
        rerr(j)=slin(k+resol)-level(j);

    elseif (slin(k)<0 && slin(k+resol)<0)

        slout(j)=-1;
        lerr(j)=slin(k)+level(j);
        rerr(j)=slin(k+resol)+level(j);

    end


    %update DFE coefficients and sampling phase

    if(lerr(j)>0 && rerr(j)>0) %update DFE

        if(slout(j)>0) level(j+1)=level(j)+steplevel/levelr;
        else level(j+1)=level(j)-steplevel;
        end

        centertemp(j+1)=centertemp(j);
        w1(j+1)=w1(j)+step1*slout(j-1);
        w2(j+1)=w2(j)+step1*slout(j-2);
        w3(j+1)=w3(j)+step1*slout(j-3);
        w4(j+1)=w4(j)+step1*slout(j-4);
        w5(j+1)=w5(j)+step1*slout(j-5);
```

```
    elseif(lerr(j)<0 && rerr(j)<0) %update DFE

        if(slout(j)>0) level(j+1)=level(j)-steplevel;
        else level(j+1)=level(j)+steplevel/levelr;
        end

        centertemp(j+1)=centertemp(j);
        w1(j+1)=w1(j)-step1*slout(j-1);
        w2(j+1)=w2(j)-step1*slout(j-2);
        w3(j+1)=w3(j)-step1*slout(j-3);
        w4(j+1)=w4(j)-step1*slout(j-4);
        w5(j+1)=w5(j)-step1*slout(j-5);

    %update phase
    elseif(lerr(j)>0 && rerr(j)<0) || (lerr(j)<0 && rerr(j)>0)

        level(j+1)=level(j);
        centertemp(j+1)=centertemp(j)-stepCDR*sign(slout(j))*sign(lerr(j));
        w1(j+1)=w1(j);
        w2(j+1)=w2(j);
        w3(j+1)=w3(j);
        w4(j+1)=w4(j);
        w5(j+1)=w5(j);

    else

        level(j+1)=level(j);
        centertemp(j+1)=centertemp(j);
        w1(j+1)=w1(j);
        w2(j+1)=w2(j);
        w3(j+1)=w3(j);
        w4(j+1)=w4(j);
        w5(j+1)=w5(j);

    end

    center(j+1)=round(centertemp(j+1)); %quantize sampling phase

end  %%end of feedback loop
```

```
%plot

figure(1); plot(level); hold on; plot(w1); plot(w2); plot(w3); plot(w4);
plot(w5); legend('dLev', 'w1', 'w2', 'w3', 'w4', 'w5');

figure(2); plot(centertemp); hold on; plot(center); legend('centertemp',
'center');

eyediagram(rxin0(end-50000:end-10), samples); %channel output
eyediagram(slin(end-50000:end-10), samples); %DFE summer output
```

Bibliography

[1] J. D. H. Alexander, “Clock recovery from random binary signals,” Electron. Lett., vol. 11, no. 22, pp. 541–542, Dec. 1975.

[2] S. Gondi et al., “Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers,” IEEE J. Solid-State Circuits, vol. 42, no. 9, pp. 1999–2011, Sep. 2007.

[3] A. Roshan-Zamir et al., “A 56-Gb/s PAM4 receiver with low-overhead techniques for threshold and edge-based DFE FIR- and IIR-tap adaptation in 65-nm CMOS,” IEEE J. Solid-State Circuits, vol. 54, no. 3, pp. 672–684, Mar. 2019.

[4] E.-H. Chen et al., “Near-optimal equalizer and timing adaptation for I/O links using a BER-based metric,” IEEE J. Solid-State Circuits, vol. 43, no. 9, pp. 2144–2156, Sep. 2008.

[5] S. Son et al., “A 2.3-mW, 5-Gb/s low-power decision-feedback equalizer receiver front-end and its two-step, minimum bit-error-rate adaptation algorithm,” IEEE J. Solid-State Circuits, vol. 48, no. 11, pp. 2693–2704, Nov. 2013.

[6] H. Noguchi et al., “A 40-Gb/s CDR circuit with adaptive decision-point

control based on eye-opening monitor feedback," IEEE J. Solid-State Circuits, vol. 43, no. 12, pp. 2929–2938, Dec. 2008.

[7] H. Won et al., "A 28-Gb/s receiver with self-contained adaptive equalization and sampling point control using stochastic sigma-tracking eye-opening monitor," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 64, no.3, pp. 664–674, Mar. 2017.

[8] H.-Y. Joo et al., "A maximum-eye-tracking CDR with biased data-level and eye slope detector for optimal timing adaptation," in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2019, pp. 243–244.

[9] B. Zhang et al., "A 28Gb/s multistandard serial link transceiver for backplane applications in 28nm CMOS," IEEE J. Solid-State Circuits, vol. 50, no. 12, pp. 3089–3100, Dec. 2015.

[10] C. Thakkar et al., "Design techniques for a mixed-signal I/Q 32-coefficient Rx-feedforward equalizer, 100-Coefficient decision feedback equalizer in an 8 Gb/s 60GHz 65 nm LP CMOS Receiver," IEEE J. Solid-State Circuits, vol. 49, no. 11, pp. 2588–2607, Nov. 2014.

[11] J. Han et al., "Design techniques for a 60 Gb/s 173mW wireline receiver frontend in 65nm CMOS technology," IEEE J. Solid-State Circuits, vol. 51, no. 4, pp. 871–880, Apr. 2016.

[12] K.-L. J. Wong et al., "Edge and data adaptive equalization of serial-link

transceivers," IEEE J. Solid-State Circuits, vol. 43, no. 9, pp. 2157–2169, Sep. 2008.

[13] H. Wang et al., "A 21-Gb/s 87-mW transceiver with FFE/DFE/analog equalizer in 65-nm CMOS technology," IEEE J. Solid-State Circuits, vol. 45, no. 4, pp. 909–920, Apr. 2010.

[14] A. Manian et al., "A 40-Gb/s 14-mW CMOS wireline receiver," IEEE J. Solid-State Circuits, vol. 52, no. 9, pp. 2407–2421, Sep. 2017.

[15] J. L. Zerbe et al., "Equalization and clock recovery for a 2.5-10 Gb/s 2-PAM/4-PAM backplane transceiver cell," IEEE J. Solid-State Circuits, vol. 38, no. 12, pp. 2121–2130, Dec. 2003.

[16] F. Spagna et al., "A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS," in IEEE ISSCC Dig. Tech. Papers, Feb. 2010, pp. 366–368.

[17] M. Hossain et al., "DDJ-adaptive SAR TDC-based timing recovery for multilevel signaling," IEEE J. Solid-State Circuits, vol. 54, no. 10, pp. 2833–2844, Oct. 2019.

[18] S. Parikh et al., "A 32Gb/s wireline receiver with a low-frequency equalizer, CTLE and 2-tap DFE in 28nm CMOS," in IEEE ISSCC Dig. Tech. Papers, Feb. 2013, pp. 28–30.

[19] D. Murphy et al., “Phase noise in LC oscillators: a phasor-based analysis of a general result and of loaded Q,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no.6, pp. 1187–1203, Jun. 2010.

[20] M. K. Simon et al., “Exponential-type bounds on the generalized marcum Q-function with application to error probability analysis over fading channels,” IEEE Tran. Communications, vol. 48, no.3, pp. 359–366, Mar. 2000.

[21] V. Stojanovic et al., “Autonomous dual-mode (PAM2/4) serial link transceiver with adaptive equalization and data recovery,” IEEE J. Solid-State Circuits, vol. 40, no. 4, pp. 1012–1026, Apr. 2005.

[22] M. Park et al., “A 7Gb/s 9.3mW 2-tap current-integrating DFE receiver,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2007, pp. 230–232.

[23] A. E. Neyestanak et al., “A 6.0-mW 10.0-Gb/s receiver with switched-capacitor summation DFE,” IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 889–896, Apr. 2007.

[24] M. Kim, “DFE weight control for memory system using asymmetric interface environment,” M.S. thesis, Seoul National University, 2018.

[25] K.-L. J. Wong et al., “A 5-mW 6-Gb/s quarter-rate sampling receiver with a 2-tap DFE using soft decisions,” IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 881–888, Apr. 2007.

[26] J. Savoj et al., “A 10-Gb/s CMOS clock and data recovery circuit with a half-rate binary phase/frequency detector,” IEEE J. Solid-State Circuits, vol. 38, no. 1, pp. 13–21, Jan. 2003.

[27] L. Kong et al., “An inductorless 20-Gb/s CDR with high jitter tolerance,” IEEE J. Solid-State Circuits, vol. 54, no. 10, pp. 2857–2866, Oct. 2019.

[28] W. Bae et al., “A 0.36 pJ/bit, 0.025 mm2, 12.5 Gb/s forwarded-clock receiver with a stuck-free delay-locked loop and a half-bit delay line in 65-nm CMOS technology,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 63, no.9, pp. 1393–1403, Sep. 2016.

[29] J. Lee et al., "Analysis and Modeling of Bang-Bang Clock and Data Recovery Circuits," IEEE J. Solid-State Circuits, vol. 39, no. 9, pp 1571–1580, Sep. 2004.

[30] M. Ierssel et al., “A 3.2Gb/s CDR using semi-blind oversampling to achieve high jitter tolerance,” IEEE J. Solid-State Circuits, vol. 42, no. 10, pp 2224–2234, Oct. 2007.

[31] M.-J. E. Lee et al., “Jitter transfer characteristics of delay-locked loops–theories and design techniques,” IEEE J. Solid-State Circuits, vol. 38, no. 4, pp. 614–621, Apr. 2003.

[32] M. Hossain et al., “A fast-lock, jitter filtering all-digital DLL based burst-mode memory interface,” IEEE J. Solid-State Circuits, vol. 49, no. 4,

pp. 1048–1062, Apr. 2014.

[33] C. Kromer et al., “A 25-Gb/s CDR in 90-nm CMOS for high-density interconnects,” IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2921–2929, Dec. 2006.

[34] F. A. Musa et al., “Modeling and design of multilevel bang-bang CDRs in the presense of ISI and noise,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no.10, pp. 2137–2147, Oct. 2007.

[35] H.-J. Jeon et al., “A bang-bang clock and data recovery using mixed mode adaptive loop gain strategy,” IEEE J. Solid-State Circuits, vol. 48, no. 6, pp. 1398–1415, Jun. 2013.

[36] M.-J. Park et al., “Pseudo-linear analysis of bang-bang controlled timing circuits,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 6, pp. 1381–1394, Jun. 2013.

[37] J. Liang et al., “Loop gain adaptation for optimum jitter tolerance in digital CDRs,” IEEE J. Solid-State Circuits, vol. 53, no. 9, pp. 2696–2708, Sep. 2018.

[38] F. A. Musa et al., “A Baud-rate timing recovery scheme with a dual-function analog filter,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 12, pp. 1393–1397, Dec. 2006.

[39] B. Razavi et al., “Design of high-speed, low-power frequency divider and phase-locked loops in deep submicron CMOS,” IEEE J. Solid-State Circuits, vol. 30, no. 2, pp. 101–109, Feb. 1995.

[40] G. Wu et al., “A 1–16 Gb/s all-digital clock and data recovery with a wideband high-linearity phase interpolator,” IEEE Trans. Very Large Scale Integr. Syst., vol. 24, no. 7, pp. 2511–2520, Jul. 2016.

[41] J. Im et al., “40-to-56Gb/s PAM-4 receiver with 10-tap direct decision-feedback equalization in 16nm FinFET,” in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, Feb. 2017, pp. 114–115.

[42] Y. Lu et al., “Design techniques for a 66Gb/s 46mW 3-tap decision feedback equalizer in 65nm CMOS,” IEEE J. Solid-State Circuits, vol. 48, no. 12, pp. 3243–3257, Dec. 2013.

[43] J. Lee et al., “A 2.44-pJ/b 1.62–10-Gb/s receiver for next generation video interface equalizing 23-dB loss with adaptive 2-tap data DFE and 1-tap edge DFE,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no.10, pp. 1295–1299, Oct. 2018.

[44] J. Lee et al., “A 0.1pJ/b/dB 1.62-to-10.8Gb/s video interface receiver with fully adaptive equalization using un-even data level,” in Proc. Symp. VLSI Circuits, Jun. 2019, pp. 198–199.

[45] J. F. Bulzacchelli et al., “A 10-Gb/s 5-tap DFE/4-tap FFE transceiver in

90-nm CMOS technology," IEEE J. Solid-State Circuits, vol. 41, no. 12, pp. 2885–2900, Dec. 2006.

[46] J. Lee et al., "Design and comparison of three 20-Gb/s backplane transceivers for duobinary, PAM4, and NRZ data," IEEE J. Solid-State Circuits, vol. 43, no. 9, pp. 2120–2133, Sep. 2008.

[47] J. Lee et al., "Design of 56 Gb/s NRZ and PAM4 SerDes transceivers in CMOS technologies," IEEE J. Solid-State Circuits, vol. 50, no. 9, pp. 2061–2073, Sep. 2015.

[48] Y. Frans et al., "A 56-Gb/s PAM4 wireline transceiver using a 32-way time-interleaved SAR ADC in 16-nm FinFET," IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 1101–1110, Apr. 2017.

[49] G.-S. Jeong et al., "A modulo-FIR equalizer for wireline communications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no.11, pp. 42784286, Nov. 2019.

초 록

이 논문에서는 최소-비트 비트 에러율 (BER)에 대한 최대 눈크기 추적 CDR (MET-CDR)의 설계가 제안되었다. 제안 된 CDR 은 최적의 샘플링 단계를 찾기 위해 반복 절차를 가진 BER 카운터 또는 아이 모니터가 필요하지 않다. 에러 샘플러 출력에 가중치를 두어 더하여 얻은 치우친 데이터 레벨 (biased dLev) 은 사전 커서 ISI(pre-cursor ISI) 의 정보도 고려한 눈 높이 정보를 추출한다. 델타 T 만큼의 시간 차이를 둔 지점에서 작동하는 두 샘플러는 현재 눈 높이와 눈 기울기의 극성을 감지하고, 이 정보를 통해 제안하는 CDR 은 눈 기울기가 0 이되는 최대 눈 높이로 수렴한다. 측정 결과는 최대 눈 높이와 최소 BER 의 샘플링 위치가 잘 일치 함을 보여준다. 28nm CMOS 공정으로 구현된 수신기 칩은 23.5dB 의 채널 손실이 있는 상태에서 26Gb/s 에서 동작 가능하다. 0.25UI 의 아이 오프닝을 가지며, 87mW 의 파워를 소비한다.

키워드: 비트 오류율, 클록 및 데이터 복구, 디시전 피드백 이퀄라이저, 고속 링크, 사전 커서 인터 심볼 간섭, 샘플링 포인트 제어, SS-LMS 알고리즘, 타이밍 적응.

학 번: 2016-30218