electronics

MDPI

Article

Analysis and Modeling of Mueller–Muller Clock and Data Recovery Circuits

Tao Liu [1], Tiejun Li [2,*], Fangxu Lv [1,2], Bin Liang [2], Xuqiang Zheng [3], Heming Wang [1], Miaomiao Wu [1], Dechao Lu [1] and Feng Zhao [4,*]

[1] School of Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; liutao199702@163.com (T.L.); lvfangxu1988@163.com (F.L.); 13992033669@139.com (H.W.); Wu_Miaomaio1995@163.com (M.W.); ludechao0954@163.com (D.L.)
[2] School of Computer, National University of Defense Technology, Changsha 410003, China; bliang@nudt.edu.cn
[3] Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China; zhengxuqiang@ime.ac.cn
[4] School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China
* Correspondence: tjli@nudt.edu.cn (T.L.); fzhao956@ustc.edu.cn (F.Z.)

Abstract: In this paper, an accurate linear model of the Mueller–Muller phase detector (MMPD)-based clock and data recovery circuit (MM-CDR) is proposed, which analyzes several critical points of the MM-CDR including the linearization of the MMPD and the gain of the voter. Using our technique, the jitter between the recovery clock and the input data can be estimated with a sub-picosecond accuracy, as demonstrated in the simulation results of a 56 Gb/s quarter-rate MM-CDR implemented in 28 nm CMOS.

Keywords: clock and data recovery; Mueller–Muller phase detector; jitter; jitter tolerance

Citation: Liu, T.; Li, T.; Lv, F.; Liang, B.; Zheng, X.; Wang, H.; Wu, M.; Lu, D.; Zhao, F. Analysis and Modeling of Mueller–Muller Clock and Data Recovery Circuits. *Electronics* **2021**, *10*, 1888. https://doi.org/10.3390/electronics10161888

Academic Editors: Anna Richelli and Fabian Khateb

Received: 24 June 2021
Accepted: 3 August 2021
Published: 6 August 2021

1. Introduction

With the rapid development of integrated circuits and the emergence of advanced process nodes, the data transfer rate of high-speed serial interfaces (i.e., serializer/deserializer) has grown exponentially [1]. The rapid growth of data rate makes the classical Bang-Bang phase detector (BBPD) no longer suitable for high-speed clock and data recovery (CDR). However, the advantage of the Mueller–Muller phase detector (MMPD) that requires only one sample per symbol alleviates the problem of BBPD timing tension and exponential growth of power consumption in high-speed situations, making Mueller–Muller baud-rate sampling widely used in the serial IO design [2–6].

Compared to the Mueller–Muller baud-rate sampling technique, a lot of work has been done on Bang-Bang CDR. In [7], Jee et al. linearized the BBPD using the effect of metastability and input jitter, combining with the behavioral characteristics of large signals to characterize the jitter transfer and jitter tolerance of the CDR. However, for jitter transfer, we are more concerned about the bandwidth of the CDR after locking (i.e., in a small range). In [8], Sonntag et al. described a linear model for the general architecture of digital clock and data recovery. They linearized the BBPD using Gaussian jitter and derived the gain formula, and then obtained a linear model of the CDR in combination with the analysis of other modules of the digital CDR loop, in which the input jitter of the BBPD needs to be simulated accurately each time. The authors in [9] summarized the existing linear models of BBPD and analyzed the effect of random jitter and finite loop delay on the loop characteristics. Nevertheless, there is no analysis of linearization for the MM-CDRs in previous work, which all focus on the use of the Mueller–Muller baud-rate sampling technique. For example, Dokania et al. [10] introduced an unequalized MM-CDR, which can adjust the phase lock position according to the extracted channel response and achieve the adaptive channel response; while Choi et al. [11] presented a weight-adjusting sign-sign

Mueller–Muller clock and data recovery circuit. These studies only apply the MMPD without a systematic analysis of the MM-CDR, affecting the circuit design performance and extending the circuit design cycle. To solve these problems, we propose the gain formula of MMPD for the first time using the jitter characteristics of the input, then give the joint analysis method of voter and MMPD, and finally analyze the error jitter for the overall system. Through the above modeling, we address the theoretical gap and provided the basis for the design for MM-CDR.

In this work, we propose a linear model for a Mueller–Muller CDR. The remainder of this article is organized as follows. Section 2 discusses the working principle of MMPD and presents a linear gain model for it. Section 3 describes the gain model of the MMPD-based voter and builds a small-signal model of the MM-CDR to analyze the jitter tolerance and jitter transfer. Section 4 analyzes the error jitter between recovery clock and input data. Section 5 demonstrates the simulation results that validate the analysis of Section 4, and Section 6 draws the conclusion.

2. Mueller–Muller Phase Detector

2.1. Working Principle of MMPD

The traditional oversampling design samples the input data more than once per unit interval, which has a large frequency capture range and fast capture speed. However, as the data transfer rate increases, oversampling designs encounter performance bottlenecks in terms of clocking and power consumption, and the Mueller–Muller baud rate sampling becomes the sampling method at high speed. Compared to the classical oversampling technique (e.g., BBPD), MMPD requires only half of the sampling clock and half of the high-speed front-end sampling circuitry in the same case, which can effectively reduce the area and power consumption. Figure 1 shows the block diagram of the MMPD structure.

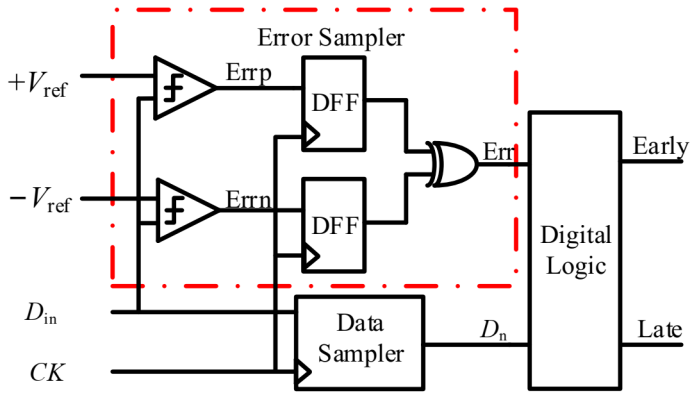

Figure 1. Mueller–Muller PD structure.

As shown in Figure 1, the MMPD mainly consists of error sampler, data sampler, and digital logic circuit [12]. The error sampler first compares the input data stream (D_{in}) with the threshold voltages "$+V_{ref}$" and "$-V_{ref}$" to generate the signals of Errp and Errn. The judgment principle is as follows. When $D_{in} > +V_{ref}$, Errp = 1; otherwise Errp = 0. When $D_{in} < -V_{ref}$, Errn = 1; otherwise Errn = 0. The signals' Errp and Errn are retimed by the recovery clock (CK) and exclusive OR to get the error information (Err). The data sampler determines the data bits (D_n) using the threshold voltage "0". Then, it utilizes the three adjacent data and error information through the digital logic circuit to produce the early/late decisions. Compared to BBPD, MMPD reduces the time axis constraint but adds two voltage thresholds to obtain the amplitude information of the sampling points (namely Errp and Errn), which detects the phase of the input data applying the amplitude information together with the waveform.

The MMPD provides updates only on data transition and requires waveform screening. As shown in Figure 2, we give four waveforms that can update the phase information,

each of which gives the early and late situation. For example, during 0-0-1 data transition, when recovering the clock lag input data, the clock sampling is denoted by green arrows, according to the previous judgment principle, the error information is ×10; when recovering the clock overrun input data, the clock sampling is represented by blue arrows, and the error information is 10×. Therefore, the early/late clock sampling information can be uniquely determined by the difference in the error information obtained during a specific data transition [13]. Based on the principles of the MMPD mentioned above, we find that the MMPD does not have a unique locking point, and will be free to wander depending on flatness of the received square wave pulse.

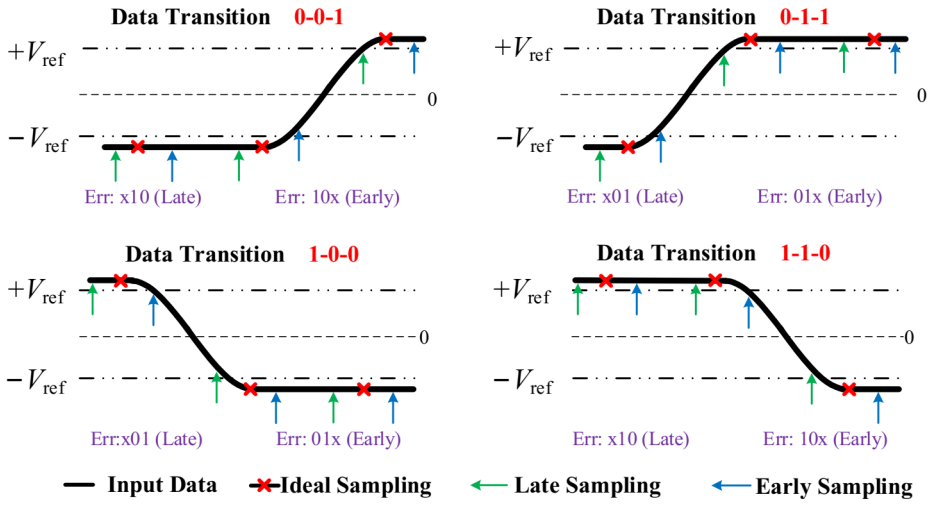

Figure 2. MMPD phase detection waveform.

Table 1 provides a truth table of the four data transition of waveforms in Figure 2. Therefore, with the Nth-order PRBS codes commonly used to simulate real data streams such as PRBS7, PRBS23, PRBS31, etc., the probability of four data transition waveforms appearing in a 2^N–1 bit cycle sequence length is 0.5. In other words, the detection density of MMPD is 0.5, which is the same as BBPD. It is worth noting that different data encoding methods may have different detection densities.

Table 1. Truth table of MMPD.

D (n − 2)	D (n − 1)	D (n)	E (n − 2)	E (n − 1)	E (n)	Phase Info.
0	0	1	x	1	0	LATE
0	0	1	1	0	x	EARLY
0	1	1	x	0	1	LATE
0	1	1	0	1	x	EARLY
1	0	0	x	0	1	LATE
1	0	0	0	1	x	EARLY
1	1	0	x	1	0	LATE
1	1	0	1	0	0	EARLY
Other						No Info.

2.2. Linear Gain of MMPD

Just as it has been done to the BBPD in [7,8,14], the binary output characteristic of the MMPD is smoothed out by the jitter inherent in the input data and the recovery clock. First, we assume that the phase error (i.e., the phase difference between the input data and the recovery clock) consists of two terms: one is the deterministic phase error $u_{e,U}$ (e.g.,

deterministic ISI or sinusoidal jitter) and the other is the random phase error $u_{e,G}$ (e.g., Gaussian white noise) [9]. That is:

$$u = \alpha \times u_{e,G} + \beta \times u_{e,U}. \tag{1}$$

where u is the average output of the MMPD, and α, β are the weights of the random phase error and the deterministic phase error, respectively.

When the phase error between the input data and the recovery clock is dominated by the random error term, we abstract the input jitter using a Gaussian probability density function with a standard deviation of σ and a mean value of 0.

Combining the operation of the MMPD in the clock and data recovery circuit, we assign "1" to phase "late" and "−1" to phase "early", and then sum the positive and negative samples with weights given by the probabilities of their occurrence. Therefore, the average output of the MMPD is:

$$u = (1)P_r(late|\phi) + (-1)P_r(early|\phi). \tag{2}$$

In Figure 3a, ϕ indicates the average phase difference between the input data and the recovery clock, $2\phi_m$ is the drift bit width, namely the part of the input data whose voltage amplitude is greater than $+V_{ref}$ or less than $-V_{ref}$. According to the Gaussian distribution probability density function, we obtain

$$P_r(late|\phi) = \frac{1}{\sigma\sqrt{2\pi}}\int_{\phi_m}^{\infty} \exp(\frac{-(x-\phi)^2}{2\sigma^2})dx. \tag{3}$$

Let $y = \frac{x-\phi}{\sqrt{2}\sigma}$ and $f(y) = e^{-y^2} \approx 1 - y^2$, then

$$P_r(late|\phi) \approx \frac{1}{2} - (\frac{\phi_m - \phi}{\sqrt{2\pi}\sigma} - \frac{(\phi_m - \phi)^3}{6\sqrt{2\pi}\sigma^3}). \tag{4}$$

Similarly, we can have

$$P_r(early|\phi) \approx \frac{1}{2} - (\frac{\phi_m + \phi}{\sqrt{2\pi}\sigma} - \frac{(\phi_m + \phi)^3}{6\sqrt{2\pi}\sigma^3}). \tag{5}$$

Substituting (4) and (5) into (2), we get

$$u_{e,G} = (1)P_r(late|\phi) + (-1)P_r(early|\phi) \approx \frac{2\sigma^2 - \phi_m^2}{\sqrt{2\pi}\sigma^3}\phi. \tag{6}$$

When the phase error between the input data and the recovery clock is dominated by the deterministic error term, we abstract the input jitter using a uniform distribution probability density function with a standard deviation of σ and a mean value of 0. From Figure 3b, we get

$$P_r(late|\phi) = \int_{\phi_m}^{\Delta+\phi} \frac{1}{2\Delta}dx = \frac{\Delta + \phi - \phi_m}{2\Delta}. \tag{7}$$

Since the variance of the uniform distribution can be expressed as

$$\sigma^2 = E[x^2] - (E[x])^2 = \frac{\Delta^2}{3}. \tag{8}$$

Substituting (8) into (7), then

$$P_r(late|\phi) = \frac{\sqrt{3}\sigma - \phi_m + \phi}{2\sqrt{3}\sigma}. \tag{9}$$

Similarly, we can attain

$$P_{\mathrm{r}}(early|\phi) = \frac{\sqrt{3}\sigma - \phi_{\mathrm{m}} - \phi}{2\sqrt{3}\sigma}. \tag{10}$$

Substituting (9) and (10) into (2), we get

$$u_{\mathrm{e,U}} = (1)P_{\mathrm{r}}(late|\phi) + (-1)P_{\mathrm{r}}(early|\phi) \approx \frac{1}{\sqrt{3}\sigma}\phi. \tag{11}$$

Note that the detection density of the input data is not considered in (6) and (11). If the detector density of the input data is λ, the gain of the MMPD based on Gaussian jitter or uniform jitter is λu.

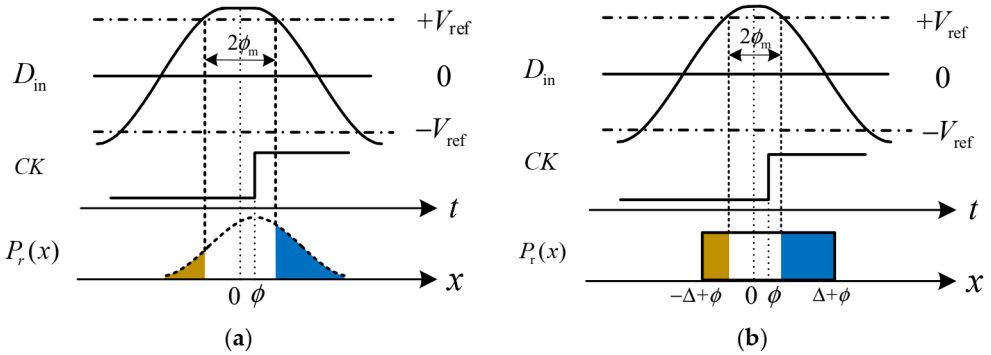

Figure 3. Smoothing of PD characteristic due to jitter. (**a**) Gaussian jitter and (**b**) uniform jitter.

By analyzing these two extreme jitter distributions of the MMPD, we perform the linearization of the MMPD and give the gain range. If the CDR system has a clean channel, then Gaussian jitter will dominate the factor and the gain of the MMPD is (6); if deterministic jitter dominates the system, the gain of the MMPD is (11). It can also be seen from (6) and (11) that the gain of the MMPD is inversely proportional to the input jitter. The higher the jitter is, the lower the gain will be. This is the key consideration when we design the MM-CDR system, because in Section 3, we will find that the performance metrics of the CDR system will vary significantly as the jitter changes.

To verify the correctness of the MMPD linear gain model, we add the corresponding jitter to the sampler clock and then scan the phase difference at the input and count the number of overruns and lags to compute its average gain. Figure 4a,b show that the theoretical calculation agrees well with the simulation results. In addition, we find that for Gaussian jitter with a jitter variance of 0.1 UI and a drift bit width of 0.05 UI, the linear region of the MMPD is about $|\phi| < 0.1$ UI; for uniform jitter with a jitter variance of 0.12 UI and a drift bit width of 0.05 UI, the linear region of the MMPD is about $|\phi| < 0.15$ UI. When we continue to increase the phase difference at the input, it can be observed that the MMPD will no longer have linear gain, but enter the nonlinear region, as illustrated in Figure 4c,d.

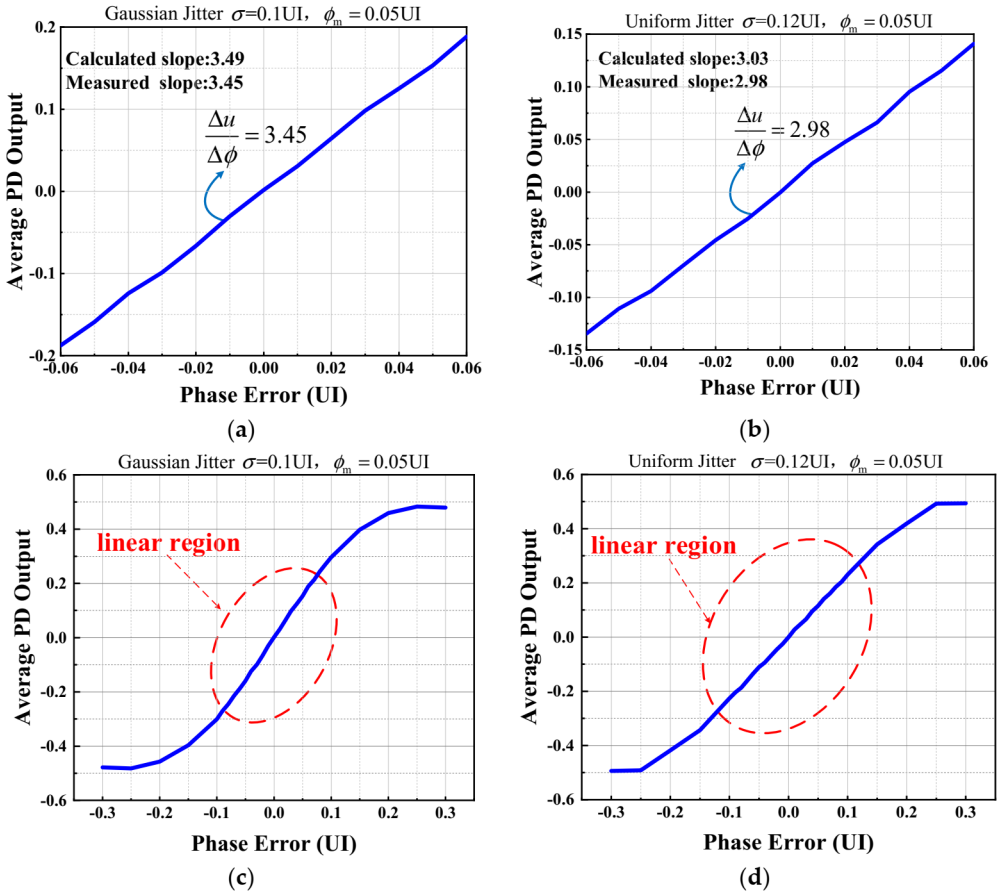

Figure 4. Simulated gain of MMPD. (**a**) Gaussian jitter at small phase difference, (**b**) uniform jitter at small phase difference, (**c**) Gaussian jitter at large phase difference, and (**d**) uniform jitter at large phase difference.

3. Small Signal Model of MM-CDR

3.1. Linear Voting Model

Majority voting is implemented in CDR by adding the N-bit parallel PD output together and then taking the sign of results. Compared to the boxcar filter, voting is able to reduce the loop delay and lower the output noise of the MMPD. Hence, the digital CDR generally adopts voting as the decimation operation.

The gain of voting relies on the MMPD output and needs to be analyzed jointly with the MMPD. Taking a bank of four Mueller–Muller phase detectors via voting as an example, we give all the possible inputs leading to the output results in Table 2.

Table 2. Input and output of vote.

Voting Output	All Circumstances					
1	1000	1100	110−1	111−1	1110	1111
−1	−1000	−1−100	−1−101	−1−1−11	−1−1−10	−1−1−1−1
0	0000	001−1	11−1−1			

Note: All inputs in the table for the voting are in no particular order.

If the input jitter is Gaussian, the probability of the MMPD output 0, −1, 1 is determined when the phase difference is fixed. According to the principle of probability statistics, the probability when the voting output is "late" (i.e., 1) is

$$P_{vote}(late) = C_4^1 P_0^3 P_{la} + C_4^2 P_0^2 P_{la}^2 + C_4^2 C_2^1 P_{la}^2 P_0 P_{ea} + C_4^1 P_{la}^3 P_{ea} + C_4^1 P_{la}^3 P_0 + P_{la}^4. \quad (12)$$

where P_0, P_{la}, and P_{ea} are the probabilities when the MMPD output are "no decision" (i.e., 0), "late" (i.e., 1), and "early" (i.e., -1) respectively.

$$P_{vote}(\text{early}) = C_4^1 P_0^3 P_{ea} + C_4^2 P_{ea}^2 P_0^2 + C_4^2 C_2^1 P_{ea}^2 P_0 P_{la} + C_4^1 P_{ea}^3 P_{la} + C_4^1 P_{ea}^3 P_0 + P_{ea}^4. \quad (13)$$

Finally, we obtain the united gain of the MMPD and the voting:

$$u = (1)P_{vote}(late) + (-1)P_{vote}(early). \quad (14)$$

To compare the impact of the voter with Boxcar on the loop and get the gain of the voter, we simulate the joint gain of MMPD and Vote, as shown by the red line in Figure 5, whose gain can be expressed as $K_{\mathrm{MD}} \times K_{\mathrm{V}}$. At the same time, we also simulate the joint gain of MMPD and boxcar filter, as depicted by the blue line in Figure 5, whose gain can be expressed as $K_{\mathrm{MD}} \times 1$. The gain ratio of these two cases can be obtained through simulation as well, and finally the gain of the voter can be derived. Figure 5a,b show that the MMPD-based voting is greatly affected by jitter and drift bit width, with the gain of the voter around 0.54 when the jitter is much larger than the drift bit width, and around 0.78 when the drift bit width is close to the jitter. Therefore, to perform analysis and fully understand the impact of voting, it is important to obtain accurate jitter and drift bit width.

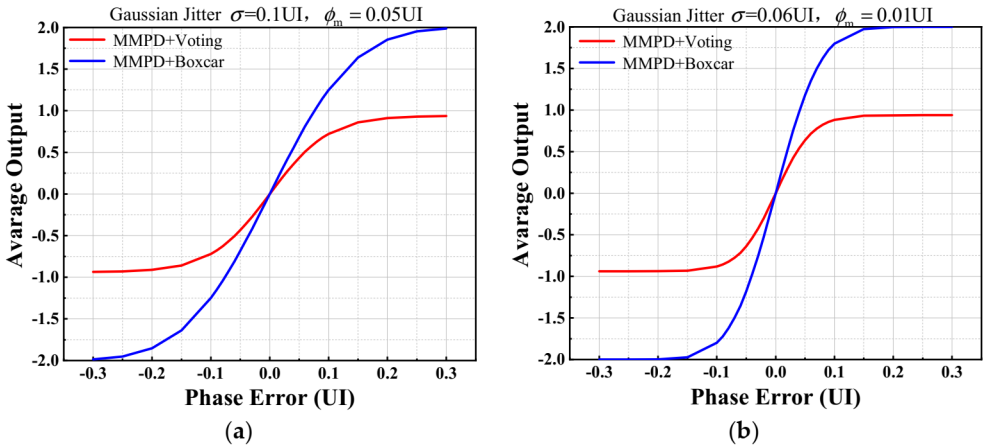

Figure 5. Simulated decimation by voting and boxcar, (**a**) $\sigma = 0.1$ UI, $\phi_m = 0.05$ UI, (**b**) $\sigma = 0.06$ UI, $\phi_m = 0.01$ UI.

3.2. Linearized Analysis of CDR System

Based on the linear model of the MMPD in Section 2.2 and the analysis of the digital CDR in [8], we can obtain the small-signal model of the MM-CDR. In this section, according to the proposed architecture in Figure 6, we first present the linearized model (see Figure 7) and then analyze its transfer function and jitter tolerance.

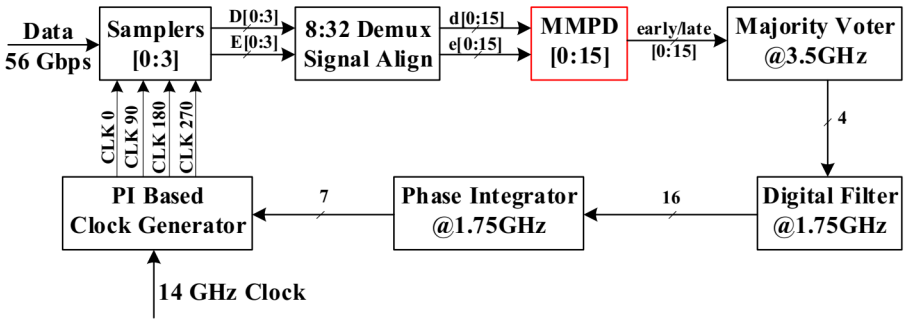

Figure 6. The architecture of 56 Gb/s quarter-rate MMPD-based digital CDR.

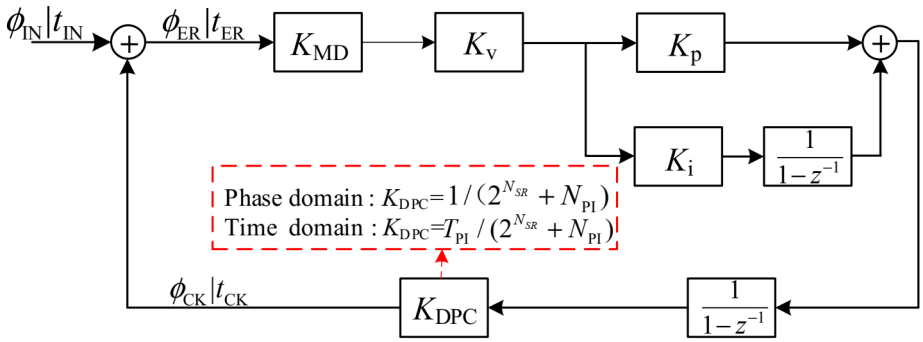

Figure 7. The linearized model of Figure 6.

Generally, the CDR must meet certain jitter performance specifications. Therefore, we first obtain the loop gain:

$$G(z^{-1}) = \frac{K_{MD}K_V K_{DPC}}{1-z^{-1}}(K_p + \frac{K_i}{1-z^{-1}}). \tag{15}$$

where K_{MD} is the gain of MMPD; K_V is the gain to handle the voting; the values of K_i and K_p correspond to the proportional and integral paths from the output of the voting to the digital-to-phase converter (DPC); the element K_{DPC} is the gain through DPC. These parameters are listed above in Table 3.

Table 3. MM-CDR parameter table.

Parameter	Value
Gain of MMPD (K_{MD})	16.0, 8.7, 5.9 per UI
Gain of vote (K_V)	$0.58 \times 32 = 19.2$
Proportional path (K_P)	1
Integral path (K_i)	2^{-13}
Phase integrator's sub-resolution (N_{SR})	6
Number of phases of the PI (N_{PI})	64

Then, the transfer function of the MM-CDR can be calculated by the following well-known equation:

$$H_{JTRAN}(z) = G(z^{-1})/(1+G(z^{-1})) \tag{16}$$

The jitter tolerance is the maximum amount of input jitter that can be tolerated by the clock and data recovery circuit at a given bit error rate (BER) [15]. Therefore, the jitter tolerance computed by (17) for a target BER is 10^{-10}.

$$H_{JTOL}(z) = (1 - \frac{12\sigma_{ER}}{T})(1+G(z^{-1})). \tag{17}$$

where σ_{ER} is the error jitter between recover clock and data, and T is the data cycle.

Figures 8 and 9 plot the jitter transfer and jitter tolerance separately. It can be seen that as the jitter increases, the bandwidth of the MM-CDR decreases and the jitter tolerance becomes smaller.

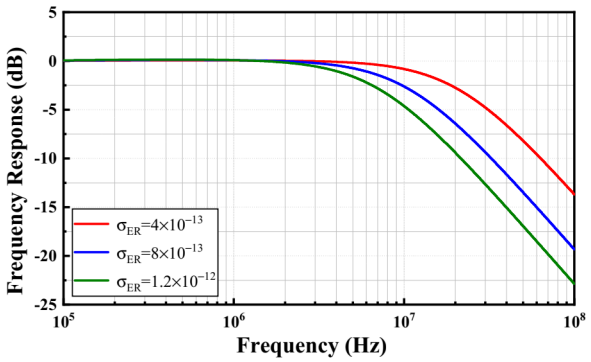

Figure 8. Jitter transfer of different input jitters.

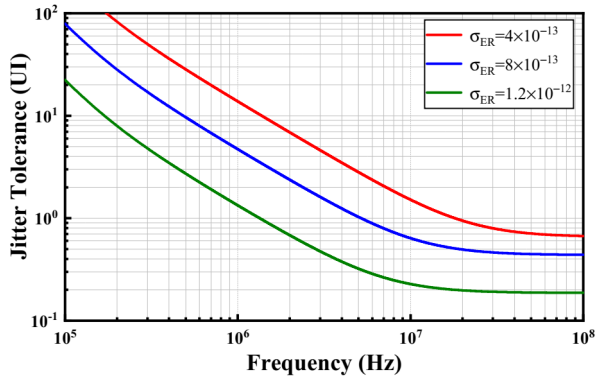

Figure 9. Jitter tolerance of different input jitters.

4. Jitter Analysis of MM-CDR

As discussed in Section 3.2 and [16], the MM-CDR loop characteristics vary with the error jitter σ_{ER}. To obtain an accurate MM-CDR model, the exact error jitter is required. One possible way is to accurately simulate the entire system at one time, measure the error jitter, and then build a linear model. Nevertheless, this will greatly lengthen the design cycle. In contrast, if we estimate the error jitter in advance based on the circuit characteristics, we can not only obtain accurate CDR performance metrics, but also shorten the design cycle. Here, we focus on the effect of the two main sources of jitter on the error jitter: input data jitter and PI quantization noise [17].

In this work, we assume that there is a clean channel and the noise of the input data is generated by the oscillator at the transmitter. For simplicity, we only consider the main noise of the oscillator (i.e., thermal noise). Its noise power spectral density (PSD) as a function of the offset frequency f is given by [18],

$$S_{\mathrm{DAT}} = \frac{n}{f^2}, \tag{18}$$

where n is the coefficient of the oscillator thermal noise. The period jitter of the input data σ_{DAT} can be obtained by integrating the phase noise with the transfer function $\left(1 - z^{-1}\right)$ [19].

$$\sigma_{\mathrm{DAT}}{}^2 = \left(\frac{T}{2\pi}\right)^2 \int_0^{\infty} \left|1 - z^{-1}\right|^2 \times S_{\mathrm{DAT}} df \approx \frac{nT^3}{2}. \tag{19}$$

Substituting (19) into (18), the noise of input data can be written as

$$S_{\mathrm{DAT}} = \frac{2\sigma_{\mathrm{DAT}}{}^2}{T^3 f^2}. \tag{20}$$

According to [20], in order to reduce the loop bandwidth and jitter peaking during design, we usually provide a larger damping factor ζ, so the transfer function can be approximated by that of a one-pole system. Using time instead of phase, the error transfer function is

$$\frac{t_{ER}}{t_{IN}} = \frac{s}{s + w_{-3dB}} \ (w_{-3dB} = K_{MD}\alpha, \alpha = \frac{K_V K_p T_{PI}}{2^{N_{SR}} N_{PI} T_{DLF}}). \tag{21}$$

where w_{-3dB} is the bandwidth of MM-CDR; T_{PI} is the period of phase interpolator's input clock; T_{DLF} is the period of digital filter. Combining (20) and (21), the error jitter contributed by the input data jitter is

$$\sigma_{ER,in}^2 = \frac{T^2}{4\pi^2} \int_0^\infty \frac{2\sigma_{DAT}^2}{T^3 f^2} \left| \frac{s}{s + w_{-3dB}} \right|^2 df = \frac{\sigma_{DAT}^2}{2T w_{-3dB}}. \tag{22}$$

According to Section 2.2, we indicate with $\sigma_{ER,in}$ the rms value of the jitter out of the MMPD, we can write:

$$K_{MD} = \lambda \frac{2\sigma_{ER,in}^2 - \phi_m^2}{\sqrt{2\pi}\sigma_{ER,in}^3}. \tag{23}$$

Note that we consider the error jitter $\sigma_{ER,in}$ caused by the input data noise as the total error jitter, and in Section 5 we will give the reasonableness analysis. Combining (21)–(23), we have,

$$\sigma_{ER,in} \approx \frac{\sigma_{DAT}^2 \sqrt{2\pi} + \sqrt{2\pi\sigma_{DAT}^4 + 32\phi_m^2 T^2 \alpha^2 \lambda^2}}{8T\alpha\lambda}. \tag{24}$$

Moreover, the quantization noise of PI is not negligible, and we consider that the PI oscillates between two phases. Assuming that the data phase has a delay of "x" with respect to the recovery clock, the phase of the recovery clock with respect to the previous moment when PI is rotated is $\Delta = T_{PI}/N_{PI}$, where Δ is the minimum step of PI. Therefore, for random jitter, "x" is continuously varying, the average error jitter is

$${\sigma_{ER,PI}}^2(x) = \frac{1}{\Delta} \int_0^\Delta \frac{(\Delta - x)^2 + x^2}{2} dx = \frac{\Delta^2}{3}. \tag{25}$$

In other words, the absolute jitter of the CDR due to quantization noise of the PI can be expressed as

$$\sigma_{ER,PI} = \frac{T_{PI}}{N_{PI}\sqrt{3}}. \tag{26}$$

The total error jitter of the system is the combination of the values of each system RJ by the root mean square summing (RSS),

$$\sigma_{ER,tot} = \sqrt{\sigma_{ER,in}^2 + \sigma_{ER,PI}^2}. \tag{27}$$

Which gives an estimate of the error jitter and will be verified in the simulation results in Section 5.

5. Results

The simulations in Figure 10 are obtained using a 56 Gb/s quarter-rate Mueller–Muller clock and data recovery circuit on 28 nm CMOS, with the structure shown in Figure 6.

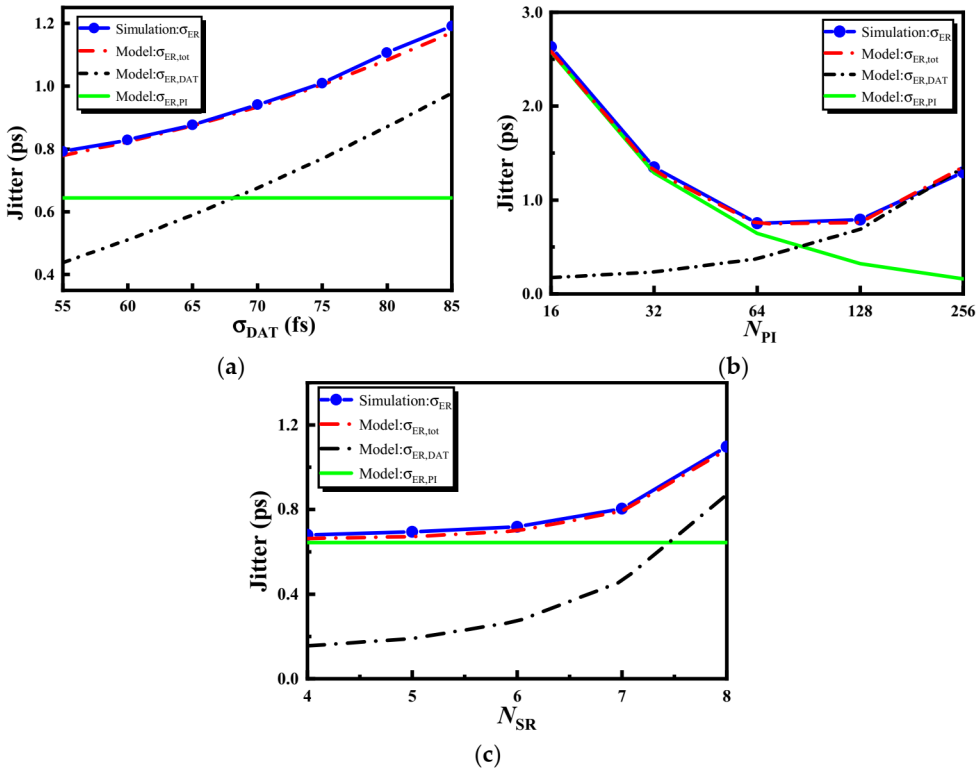

Figure 10. Comparison of model and simulation results of the error jitter with various (**a**) σ_{DAT}, (**b**) N_{PI}, and (**c**) N_{SR}.

We use a 56 GHz clock with period jitter to generate a PRBS31 code as the input data. Varying the input data jitters, the number of phases of the PI and sub-resolution of the phase integrator measure the error jitters between the recovered clock and the input data. The results are shown in Figure 10, where black dashed, green solid, and the red dashed lines correspond to (24), (26) and (27), respectively. The blue line with circles represents the error jitter of a CDR operating on 56 Gbps of serial data. Figure 10a–c respectively show the relationship between the results and the following factors: input data jitter, the number of phases of the PI, and sub-resolution of the phase integrator. it can be seen from Figure 10a that the error jitter caused by the input jitter is the dominant factor at this point, and when σ_{DAT} increases, the total error jitter increases as well.

According to Figure 10b, the total error jitter follows the quantization noise of PI when N_{PI} is low, and when N_{PI} is high, $\sigma_{\mathrm{ER,tot}}$ varies in accordance with $\sigma_{\mathrm{ER,in}}$. In fact, the MM-CDR bandwidth is large at low N_{PI}, and (21) is high-pass, thus the input jitter has little effect on the total error jitter, and the quantization noise of PI accounts for the dominant factor. This has been validated by the simulation results in Figure 10b.

In Figure 10c, we can find that although the contribution of the input jitter to the total error jitter increases as N_{SR} increases, it is still too small compared to the quantization noise of PI, and the fluctuation of $\sigma_{\mathrm{ER,in}}$ does not cause the growth of the total error jitter. When $\sigma_{\mathrm{ER,in}}$ is the dominant factor, the total error jitter approaches the error jitter caused by the input jitter, so the estimate using (23) is reasonable.

6. Conclusions

In this paper, we present a linear model of the MM-CDR. First, we design a model for the MMPD gain as in Equation (6) when Gaussian jitter is the dominant factor and Equation (11) when uniform jitter is the dominant one. In order to build a complete linear model, we introduce an estimation model for the error jitter, which achieves a sub-picosecond accuracy in the high-speed case. It is worth mentioning that the gain of the

MMPD-based voter varies greatly with the error jitter and drift bit width, so it needs to be simulated accurately.

Author Contributions: Conceptualization, T.L. (Tao Liu) and T.L. (Tiejun Li); methodology, T.L. (Tao Liu); validation, T.L. (Tao Liu), M.W.; D.L.; formal analysis, T.L. (Tao Liu), F.L.; investigation, M.W., D.L.; resources, T.L. (Tiejun Li), H.W., B.L.; data curation, H.W.; writing—original draft preparation, T.L. (Tao Liu), F.Z.; writing—review and editing, T.L. (Tao Liu), X.Z., F.L.; supervision, B.L., T.L. (Tiejun Li), H.W., F.Z.; project administration, T.L. (Tiejun Li); funding acquisition, X.Z., F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China under Grant 2018YFB2202300.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shin, D.; Kim, H.S.; Liu, C.c.; Wali, P.; Murthy, S.K.; Fan, Y. A 23.9-to-29.4GHz Digital LC-PLL with a Coupled Frequency Doubler for Wireline Applications in 10 nm FinFET. In Proceedings of the 2021 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 13–22 February 2021; pp. 188–190.
2. Rahman, W.; Yoo, D.; Liang, J.; Sheikholeslami, A.; Tamura, H.; Shibasaki, T.; Yamaguchi, H. A 22.5-to-32-Gb/s 3.2-pJ/b Referenceless Baud-Rate Digital CDR with DFE and CTLE in 28-nm CMOS. *IEEE J. Solid State Circuits* **2017**, *52*, 3517–3531. [CrossRef]
3. Ting, C.; Liang, J.; Sheikholeslami, A.; Kibune, M.; Tamura, H. A blind baud-rate ADC-based CDR. In Proceedings of the 2013 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, San Francisco, CA, USA, 17–21 February 2013; pp. 122–123.
4. Joy, A.K.; Mair, H.; Lee, H.; Feldman, A.; Portmann, C.; Bulman, N.; Crespo, E.C.; Hearne, P.; Huang, P.; Kerr, B.; et al. Analog-DFE-based 16Gb/s SerDes in 40nm CMOS that operates across 34dB loss channels at Nyquist with a baud rate CDR and 1.2Vpp voltage-mode driver. In Proceedings of the 2011 IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, 20–24 February 2011; pp. 350–351.
5. Harwood, M.; Warke, N.; Simpson, R.; Leslie, T.; Amerasekera, A.; Batty, S.; Colman, D.; Carr, E.; Gopinathan, V.; Hubbins, S.; et al. A 12.5Gb/s SerDes in 65nm CMOS Using a Baud-Rate ADC with Digital Receiver Equalization and Clock Recovery. In Proceedings of the 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, San Francisco, CA, USA, 11–15 February 2007; pp. 436–591.
6. Francese, P.A.; Toifl, T.; Buchmann, P.; Brändli, M.; Menolfi, C.; Kossel, M.; Morf, T.; Kull, L.; Andersen, T.M. A 16 Gb/s 3.7 mW/Gb/s 8-Tap DFE Receiver and Baud-Rate CDR With 31 kppm Tracking Bandwidth. *IEEE J. Solid State Circuits* **2014**, *49*, 2490–2502. [CrossRef]
7. Jri, L.; Kundert, K.S.; Razavi, B. Analysis and modeling of bang-bang clock and data recovery circuits. *IEEE J. Solid State Circuits* **2004**, *39*, 1571–1580. [CrossRef]
8. Sonntag, J.L.; Stonick, J. A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links. *IEEE J. Solid State Circuits* **2006**, *41*, 1867–1875. [CrossRef]
9. Park, M.; Kim, J. Pseudo-Linear Analysis of Bang-Bang Controlled Timing Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2013**, *60*, 1381–1394. [CrossRef]
10. Dokania, R.; Kern, A.; He, M.; Faust, A.; Tseng, R.; Weaver, S.; Yu, K.; Bil, C.; Liang, T.; Mahony, F.O. A 5.9pJ/b 10Gb/s serial link with unequalized MM-CDR in 14nm tri-gate CMOS. In Proceedings of the 2015 IEEE International Solid-State Circuits Conference—(ISSCC). Digest of Technical Papers, San Francisco, CA, USA, 22–26 February 2015; pp. 1–3.
11. Choi, M.C.; Ko, H.G.; Oh, J.; Joo, H.Y.; Lee, K.; Jeong, D.K. A 0.1-pJ/b/dB 28-Gb/s Maximum-Eye Tracking, Weight-Adjusting MM CDR and Adaptive DFE with Single Shared Error Sampler. In Proceedings of the 2020 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA, 16–19 June 2020; pp. 1–2.
12. Spagna, F.; Chen, L.; Deshpande, M.; Fan, Y.; Gambetta, D.; Gowder, S.; Iyer, S.; Kumar, R.; Kwok, P.; Krishnamurthy, R.; et al. A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS. In Proceedings of the 2010 IEEE International Solid-State Circuits Conference—(ISSCC), San Francisco, CA, USA, 7–11 February 2010; pp. 366–367.
13. Maity, S.; Mehrotra, P.; Sen, S. An Improved Update Rate Baud Rate CDR for Integrating Human Body Communication Receiver. In Proceedings of the 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), Cleveland, OH, USA, 17–19 October 2018; pp. 1–4.
14. Ramezani, M.; Andre, C.; Salama, T. Jitter analysis of a PLL-based CDR with a bang-bang phase detector. In Proceedings of the the 2002 45th Midwest Symposium on Circuits and Systems, Tulsa, OK, USA, 4–7 August 2002; p. III.
15. Talegaonkar, M.; Inti, R.; Hanumolu, P.K. Digital clock and data recovery circuit design: Challenges and tradeoffs. In Proceedings of the 2011 IEEE Custom Integrated Circuits Conference (CICC), San Jose, CA, USA, 19–21 September 2011; pp. 1–8.
16. Liang, J.; Sheikholeslami, A.; Tamura, H.; Yamaguchi, H. On-Chip Jitter Measurement Using Jitter Injection in a 28 Gb/s PI-Based CDR. *IEEE J. Solid State Circuits* **2018**, *53*, 750–761. [CrossRef]

17. Liang, J. Loop Gain Adaptation for Optimum Jitter Tolerance in Digital CDRs. *IEEE J. Solid State Circuits* **2018**, *53*, 2696–2708. [CrossRef]
18. Behzad, R. Predicting the Phase Noise and Jitter of PLLBased Frequency Synthesizers. In *Phase-Locking in High-Performance Systems: From Devices to Architectures*; IEEE: New York, NY, USA, 2003; pp. 46–69.
19. Azeredo-Leme, C. Clock Jitter Effects on Sampling: A Tutorial. *IEEE Circuits Syst. Mag.* **2011**, *11*, 26–37. [CrossRef]
20. Razavi, B. *Design of Integrated Circuits for Optical Communications*; John Wiley & Sons: Hoboken, NJ, USA, 2012.