Appendix A
Jitter Analysis

Bit error rate (BER) represents the quality of the communication in the communication system. It means the proportion of error occurrence while receiver receives data from transmitter. The reason of error occurrence is due to bad transmitter performance, cockamamie quality of channel, or bad receiver performance. Moreover, the main reason of bad receiver performance results from data jitter and clock jitter.

In this appendix, we will analyze the influence of jitter on our proposed CDR.

We discuss this issue into two parts. Part one introduces that the derived bit error rate due to the total jitter in our CDR influences the whole CDR performance while part two introduces that the maximum jitter tolerance performance in our CDR system with different zero block does not results in error. In addition, jitter transfer and jitter generation is also introduced.

According to the origin, jitter can be separated into two kinds: Random jitter (RJ) and deterministic jitter (DJ). Random jitter is represented as unbounded components and usually expressed its statistical behavior by Gaussian distribution. Oppositely, deterministic jitter is represented as bounded components and cannot be expressed by Gaussian distribution. Random jitter originates from thermal noise of the electronic devices while deterministic jitter originates from pattern jitter, Inter-Symbol Interference (ISI), or duty cycle distortion. Generally speaking, random jitter is represented by root-mean-square (RMS) value or standard variation while deterministic jitter is represented by peak-to-peak value. When we want to combine this two kinds of jitter to calculate the total jitter, we should do convolution with the probability density function (PDF) of random jitter and deterministic jitter instead of adding two values directly.

A.1 Bit Error Rate Analysis

First, we represent several non-ideal effects in our CDR architecture in Fig. A-1. Because of integrating two-stage limiting amplifier (LA) in the front of CDR loop, received data is amplified to digital signal. Amplitude noise and voltage variation eliminate and can be neglected. Thus, we analyze the bit error rate without considering only the timing error, or jitter. We can briefly observe our CDR suffers some jitter, such as data jitter (T_{jd}), clock jitter (T_{jc}), and sampling timing offset (T_{os}). We can define the timing margin (T_{margin}) as:

$$T_{margin} = \frac{T_{bit}}{2} - T_{os} - T_{jc} - T_{jd} \tag{A-1}$$

The larger the timing margin (T_{margin}), the smaller error probability of sampled data, and vice versa. Sampling timing offset (T_{os}) represents the phase offset between sampling phase and the center of the data. The larger the sampling timing offset, the smaller error occurrence. Data jitter (T_{jd}) depends on the performance of transmitter, channel, and receiver while clock jitter (T_{jc}) depends on the performance of phase-locked loop. Data jitter consists of both random jitter and deterministic jitter while clock jitter is composed of only random jitter. If the timing margin T_{margin} decreases due to these jitter, bit error rate will increase.

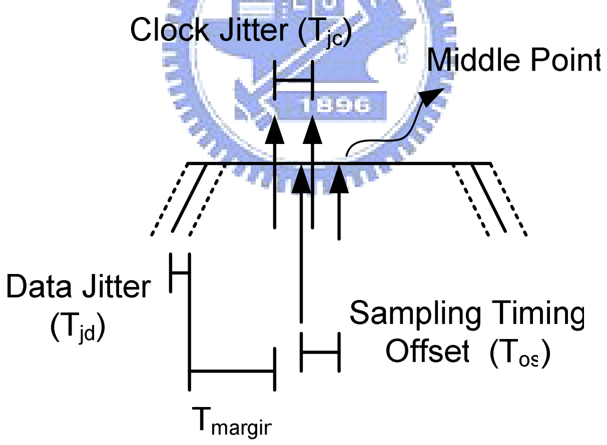


Fig. A-1 All Kinds of Jitter in Whole CDR System

The sampling timing offset T_{os} relates to the number of sampling phase in a bit time, or unit interval (UI). It also means the step number of our phase interpolator (PI) and called the resolution of the phase interpolator, k. The larger the resolution of the phase interpolator k, the smaller the sampling timing offset T_{os} and the smaller bit error rate. However, if k is larger, the more hardware complexity will be needed and power consumption and occupied area will be larger. That is a significant trade-off.

We discuss the influence of sampling timing offset on bit error rate first. The worst case of sampling data is that the center point of data falls on the center of optimum two sampling phases. We define as the maximum sampling timing offset $T_{os,max}$:

$$T_{os,\max} = \frac{T_{bit}}{2 \times k} \tag{A-2}$$

where k is the resolution of the phase interpolator. (A-2) is the expression while k is odd. If the center point of data does not fall on the center of optimum two sampling phases, CDR system will choose the sampling phase which is nearer the data center. If data rate in transmitter and receiver has some frequency offset, the value of T_{os} will be between 0 and T_{bit}/2k and its probability density function (PDF) is uniform distribution.

First, we discuss the influence of random jitter of the input data on bit error rate. we define random jitter as a Gaussian distribution with given mean and variance σ_d and normalize the bit time to one. Therefore, the bit error rate can be expressed as:

$$LBER = TD \times \int_{st}^{\infty} \frac{1}{\sqrt{2\pi\sigma_d^{\,2}}} \exp(-\frac{t^2}{2\sigma_d^{\,2}})dt \tag{A-3}$$

$$RBER = TD \times \int_{-\infty}^{st} \frac{1}{\sqrt{2\pi\sigma_d^{\,2}}} \exp(-\frac{(t-1)^2}{2\sigma_d^{\,2}})dt \tag{A-4}$$

$$TBER = LBER + RBER \tag{A-5}$$

where LBER and RBER represents the bit error rate caused by random jitter of the left-side boundary and the right-side boundary of the input data, respectively. TBER represents the total bit error rate caused by random jitter of the input data. St is the sample point and TD is the data transition density and yields a value of 1 for a

“1010 ….” and 1/2 for a PRBS pattern.

Next, we discuss deterministic jitter of the input data. The probability density function (PDF) of deterministic jitter can be defined as a dirac function (impulse response) in order to briefly analyze its characteristic. Therefore, the PDF of deterministic jitter can be expressed as:

$$DJ = a1 * \delta(t - w1) + a2 * \delta(t + w2) \qquad \text{(A-6)}$$

where w is deterministic peak-to-peak jitter and a1 and a2 are the right-side and left-side probability of dirac function, where a1+a2=1.

We can obtain the PDF of the input data jitter ($\mathrm{PDF}_{\mathrm{data}}$) by doing convolution with the PDF of deterministic jitter and random jitter, as illustrated in Fig. A-2.

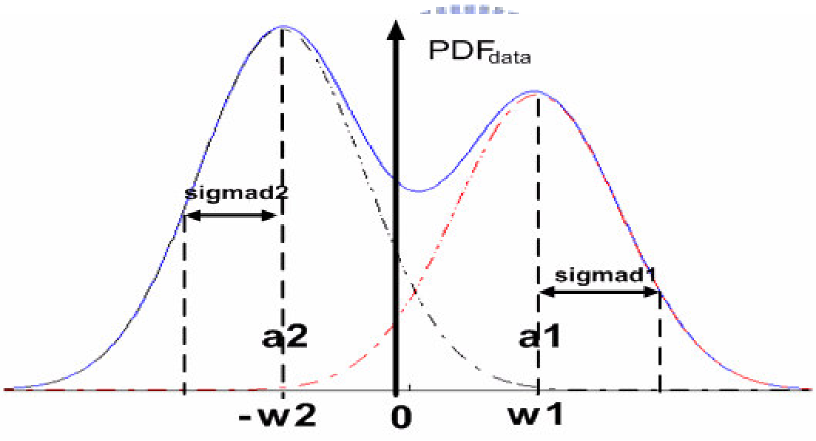


Fig. A-2 The Probability Density Function of the Input Data Jitter

After analyzing the influence of input data jitter on bit error rate, we discuss the influence of sampling clock jitter on bit error rate. Clock jitter can be represented as a random jitter. Thus, we describe the PDF of clock jitter ($\mathrm{PDF}_{\mathrm{clk}}$) as a Gaussian distribution and it is shown as:

$$PDF_{clk}(st, \sigma_c) = \frac{1}{\sqrt{2\pi\sigma_c^{\ 2}}} e^{-\frac{(st-0.5)^2}{2\sigma_c}} \qquad \text{(A-7)}$$

Furthermore, the sampling timing offset T_{os} which is caused by clock sampling also affects on bit error rate. The PDF of the sampling timing offset T_{os} ($\mathrm{PDF}_{\mathrm{os}}$) can be expressed as a Uniform distribution between 0 and 1/2k and shown as:

$$PDF_{os}(t,k)=\{\begin{matrix} k,-1/2k<t<1/2k \\ 0, \qquad others \end{matrix} \tag{A-8}$$

As the same as before, we can combine (A-7) and (A-8) by doing convolution with them . Then we express the PDF as:

$$u(st,k,\sigma_c)=PDF_{clk}*PDF_{os}=k\int_{\frac{-1}{2k}}^{\frac{1}{2k}}\frac{1}{\sqrt{2\pi}\sigma_c}e\frac{-(st-t-0.5)^2}{2\sigma_c{}^2}dt \tag{A-9}$$

Finally, after combining (A-3), (A-4), (A-6), and (A-9), we can calculate the total bit error rate, as shown below:

$$LBER=\frac{TD}{2}\cdot\int_{-\infty}^{\infty}\int_{st}^{\infty}\frac{1}{\sqrt{2\pi}}\left(\frac{a1}{\sigma_{d1}}e^{-\frac{(x-w1)^2}{2\sigma_{d1}^2}}+\frac{a2}{\sigma_{d2}}e^{-\frac{(x-w2)^2}{2\sigma_{d2}^2}}\right)u(st)dxdst \tag{A-10}$$

$$RBER=\frac{TD}{2}\cdot\int_{-\infty}^{\infty}\int_{-\infty}^{st}\frac{1}{\sqrt{2\pi}}\left(\frac{a1}{\sigma_{d1}}e^{-\frac{(x-1-w1)^2}{2\sigma_{d1}^2}}+\frac{a2}{\sigma_{d2}}e^{-\frac{(x-1-w2)^2}{2\sigma_{d2}^2}}\right)u(st)dxdst \tag{A-11}$$

$$TBER=LBER+RBER \tag{A-12}$$

We observe (A-10) and (A-11) preciously, where *u(st)* is derived from (A-9). The most important parameter in (A-9), which is explained the influence of sampling timing offset T_{os} on the total bit error rate (TBER), is phase interpolator resolution (k). For a specific transmission environment and a local clock generator with given performance, clock jitter and data jitter in the whole CDR system is almost fixed. It implies that phase interpolator resolution (k) dominates the total bit error rate. Thus, for a given and fixed data jitter and clock jitter, for example, a Gaussian distribution with 0.1UI variance respectively, we can describe the relationship between bit error rate BER and phase interpolator resolution (k), as shown in Fig. A-3. As our expectation, if phase interpolator resolution k is higher, bit error rate BER will be decrease. However, the trend approaches to saturation. In order to achieve the total bit error rate less than 10^{-12} and make reservation for some margin, we design k=32 for our CDR system. Therefore, when CDR is in the lock condition, the static phase offset, or called static phase error, will be less than 1/32UI, as mentioned in section 3.1. If we choose k=64, more hardware complexity and more power consumption is inevitable.

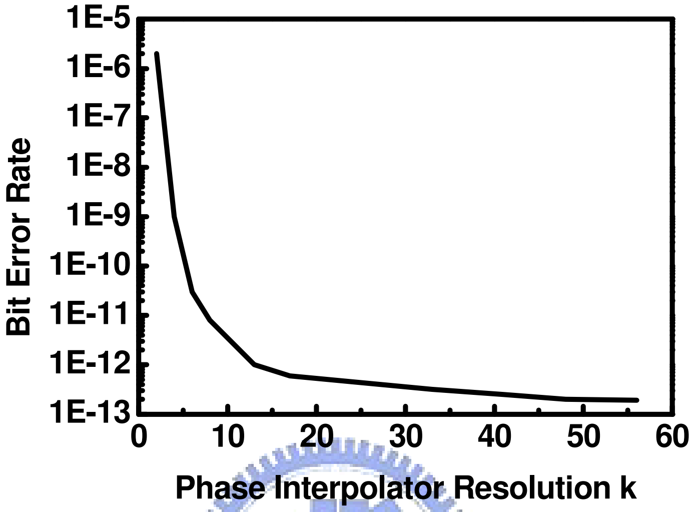


Fig. A-3 Bit Error Rate (BER) versus Phase Interpolator Resolution (k)

A.2 Jitter Tolerance

In this section, we discuss our CDR system can stand for how much jitter until error occurrence. We assume that the frequency of noise varies in the form of sinusoidal waveform and this jitter frequency modulates the input data rate by frequency modulation (FM), as illustrated in Fig. A-4. When the receiver receives sequential ones or zeros, CDR cannot change sampling phase because there is no data transition. CDR could make a mistake in this interval because data rate is affected by noise modulation.

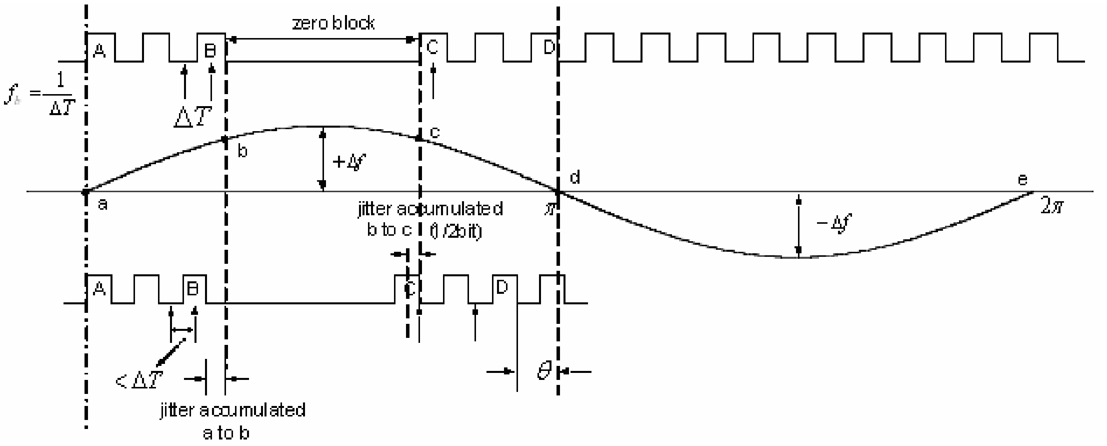


Fig. A-4 The Jitter Frequency modulates the Data Rate (I)

As shown in Fig. A-4, the top signal is the data which is not interfered by noise, the middle signal is noise, and the nethermost is the data which is modulated by jitter. In the interval a to d, the jitter frequency variation is positive and data rate becomes higher after data is modulated by jitter. On the other hand, in the interval d to e, the jitter frequency variation is negative and data rate becomes slower.

In the interval b to c, the transmitter transmits a stream of zero, called “zero block” or “zero run”. There is no data transition, so CDR cannot modify the sampling phase. For this reason, phase detector or sampler in CDR could sample the last data or the next data and CDR makes a mistake. In the interval a to b, even if data rate always changes, there is theoretically no error occurrence as long as frequency variation is not too fast. That is because the sampling phase always changes along with data rate. Hence, the sampling phase always falls on the data center . In the general condition, the sampling phase should fall on the data center when sampling the data C. Nevertheless, because there is no data transition in the interval b to c, the sampling phase cannot be modified and error could occur. If the timing margin of the data is over 1/2UI, the sampling phase will fall on another data.

After learning the above concepts, we start to calculate jitter tolerance. We separate into two conditions to discuss. The first condition is when the jitter frequency is just a little slower than the data rate or almost the same. Because there is no data transition in the half period where jitter frequency always varies, the error occurs if the peak-to-peak jitter is more than 1/2UI. Thus, in this condition, jitter tolerance is 1/2UI.

The second condition is when the jitter frequency is much slower than the data rate. Fig. A-5 describes the variation of data rate after modulated by jitter frequency. f_b is the original data rate and f_j is the jitter frequency. x represents the bit number of data among 0 to π while n represents the zero block number. The gray part is the region where is no data transition.

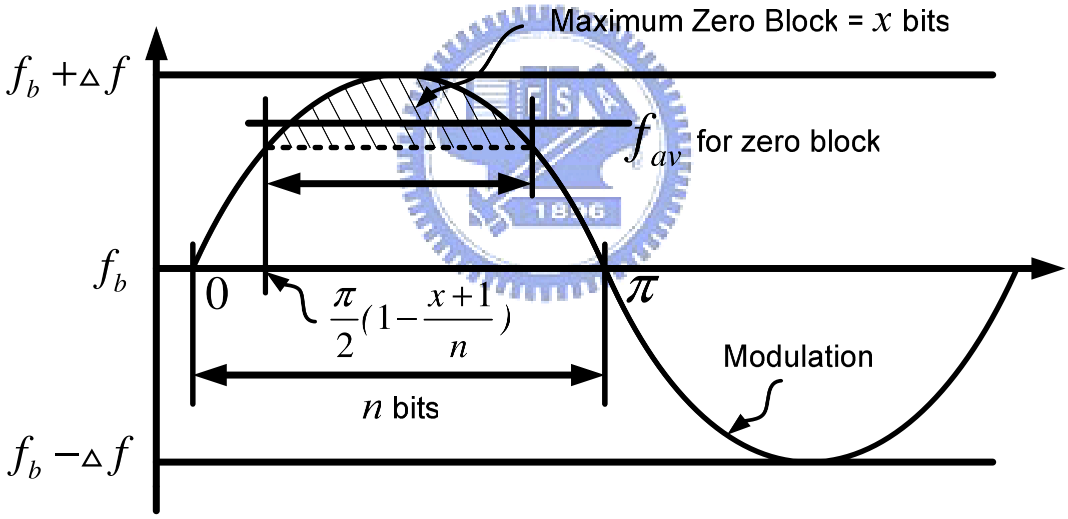


Fig. A-5 The Jitter Frequency modulates the Data Rate (II)

The data rate, which is modulated by the jitter frequency, can be expressed as:

$$Data \quad Rate = f_b - \Delta f \sin\theta \qquad \text{(A-12)}$$

If we integral (A-12), the result will be the average frequency of the period where is no data transition. Then we divide the result by (n+1)/x because there are n+1 bits in this region. CDR changes the sampling phase according to the (n+1)th bit which has data transition information until the (n+2)th bit. Hence, f_{av} represents the average frequency of the period where is no data transition and expressed as:

$$\begin{aligned} f_{av} &= f_b - \Delta f \cdot 2\int_{\pi/2[1-(n+1)/x]}^{\pi/2} \frac{sin\theta d\theta}{(n+1)/x} \\ &= f_b - \frac{x}{(n+1)} \cdot \Delta f \cdot 2\int_{\pi/2[1-(n+1)/x]}^{\pi/2} sin\theta d\theta \end{aligned} \qquad \text{(A-13)}$$

$f_b - f_{av}$ represents the average jitter of every bit of the period where is no data transition. Then we time it by (n+1) / f_b to express the total accumulated jitter in this interval. As mentioned before, error occurrence while the accumulated jitter is more than 1/2UI where is no data transition. Thus, we express as:

$$\frac{1}{2} = (f_b - f_{av}) \cdot \frac{n+1}{f_b} \qquad \text{(A-14)}$$

We put f_{av} in (A-13) to (A-14) and obtain that:

$$\frac{1}{2} = \frac{2\Delta f x}{\pi f_b} \cos\frac{\pi}{2}(1 - \frac{n+1}{x}) \qquad \text{(A-15)}$$

Then we derive the peak-to-peak jitter tolerance. The jitter accumulated in the interval a to d is the peak-to-peak jitter. Therefore, we calculate the average frequency of every bit among 0 to π in the interval a to d and denote as f_{av2}:

$$f_{av2} = f_b - \Delta f \cdot \frac{1}{\pi}\int_0^{\pi} \sin\theta d\theta = f_b - \frac{2}{\pi}\Delta f \qquad \text{(A-16)}$$

There are x bits in the interval a to d. Then we time $f_b - f_{av2}$ by x/ f_b to represent the jitter accumulated in the interval a to d, and it is also the peak-to-peak jitter, shown as:

$$J_{pp} = (f_b - f_{av2}) \cdot \frac{x}{f_b} \qquad \text{(A-17)}$$

We put f_{av2} in (A-16) to (A-17) and obtain that:

$$J_{pp} = \frac{2}{\pi}\Delta f \cdot \frac{x}{f_b} \qquad \text{(A-18)}$$

Then we put Δf in (A-15) to (A-18) and obtain the maximum jitter tolerance, describe as:

$$J_{pp} = \frac{1}{2} \cdot \frac{1}{\cos(\pi/2)[1-(n+1)/m]}$$

$$\Delta f = \frac{\pi}{4} \frac{f_b}{x\cos(\pi/2)[1-(n+1)/x]} \qquad \text{(A-19)}$$

where we can also obtain the amplitude of jitter frequency variation Δf, and $m = \frac{f_b}{2f_j}$. (A-19) comes into existence while $\frac{n+1}{m} \leq 1$. That is the condition when the jitter frequency is much slower than the data rate.

On the other hand, while $\frac{n+1}{m} \geq 1$, that is the condition when the jitter frequency is just a little slower than the data rate or almost the same, the maximum jitter tolerance describes as:

$$J_{pp} = \frac{1}{2} \quad for \frac{n+1}{m} \geq 1$$

$$\Delta f = \frac{\pi}{4} \cdot f_j \quad for \frac{n+1}{m} \geq 1 \qquad \text{(A-20)}$$

Because our CDR is quarter-rate architecture, clock generated by phase-locked loop is one-fourth the data rate, the maximum jitter tolerance is also one-fourth of the full-rate CDR architecture. Thus, peak-to-peak jitter shown in (A-19) and (A-20) must be modified as:

$$J_{pp} = \frac{1}{8} \cdot \frac{1}{cos(\pi/2)[1-(n+1)/x]} \quad for \frac{n+1}{m} \leq 1$$
$$J_{pp} = \frac{1}{8} \quad for \frac{n+1}{m} \geq 1 \tag{A-21}$$

The most important parameter in (A-21) is *n*, which represents the zero block number. We should consider jitter tolerance and maximum zero block tolerance simultaneously. Although there is no specification limitation on jitter tolerance for burst-mode PON system, we follow the specification for OC-48. We must ensure that how many numbers of zero block can be tolerated while our CDR passes the OC-48 mask. As shown in Fig. A-6, the more numbers of zero block tolerance, the less jitter can be tolerated. It implies a trade-off. By simulation result, we can observe while our CDR passes the OC-48 mask, the maximum zero block tolerance is 17 bits.

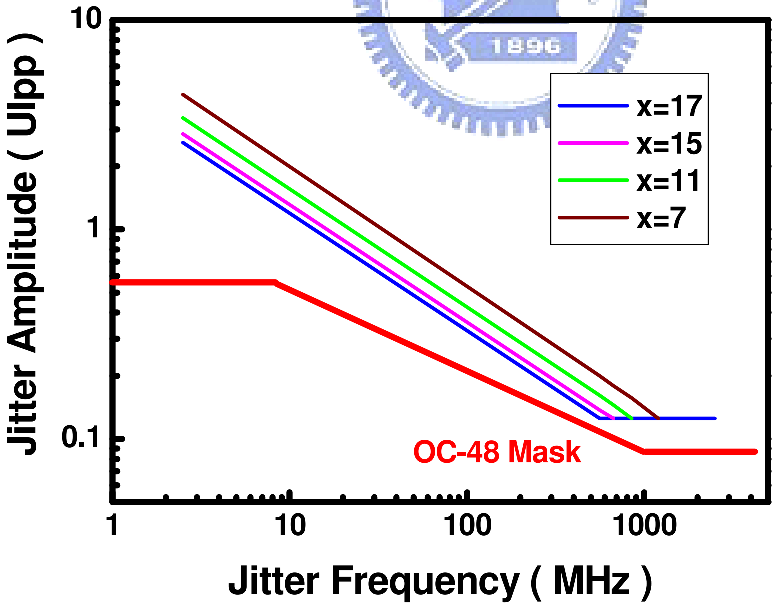


Fig. A-6 Jitter Tolerance with Different Zero Block Tolerance

A.3 Jitter Transfer

Jitter transfer means the output data jitter as the input data jitter is varied at different rates, so it presents the capacity of jitter filtering. For data repeaters, because they are connected in series, jitter transfer is an important specification to filter out accumulated jitter on the path, if not, after tens of repeaters, the signal must have been overwhelmed by a variety of noise sources.

What is the desirable jitter transfer function for CDR circuits? We note that if the input jitter varies very slowly, for example, if the zero crossings wander from their ideal points at a low rate, than the output must follow the input to ensure phase-locking. On the other hand, if the input jitter varies rapidly, the CDR circuit must filter out the jitter, for example, the output must track the input to a less extent. Thus, the jitter transfer exhibits a low-pass characteristic, as is the case with phase-locked loop.

Unlike jitter transfer, jitter tolerance, which is mentioned in the last section, is a measurement of tolerable jitter amplitude at a given frequency of jitter modulation. Hence, the faster the phase of recovered clock could be adjusted, the better the jitter tolerance would be. In summary, jitter tolerance is a test for phase tracking ability.

Compared jitter tolerance with jitter transfer, the slower the phase adjustment, the better the jitter filtering. It implies that jitter tolerance and jitter transfer form a trade-off problem. Faster tracking leads to better jitter tolerance but worse jitter transfer while slower tracking leads to better jitter transfer but worse jitter tolerance.

A.4 Jitter Generation

Jitter generation means the output jitter when there is no input jitter. In other words, the extra jitter is generated by the CDR system. This characteristic directly affects the timing margin of following circuits, so it is also important to a CDR system even though it is not particularly defined in the specification of PON system. This jitter stems from noise in the CDR itself, such as phase noise in VCO or supply noise.

Jitter generation is smaller in our CDR system theoretically because it operates in a closed-loop way. This closed tracking loop can filter out not only input jitter but also the noise in this CDR circuit. Besides, unlike conventional PLL-based CDR topology or Gated-VCO-based CDR topology, each phase adjustment is discrete-time instead of continuous-time. A few UI jitter appear at recovered clock and data while the output phase is changed each time. This is so called the static phase offset, as mentioned before. However, when our CDR is in the lock condition, the static phase offset is less than 1/32UI. This amount of jitter generation is quite small and can be neglected.

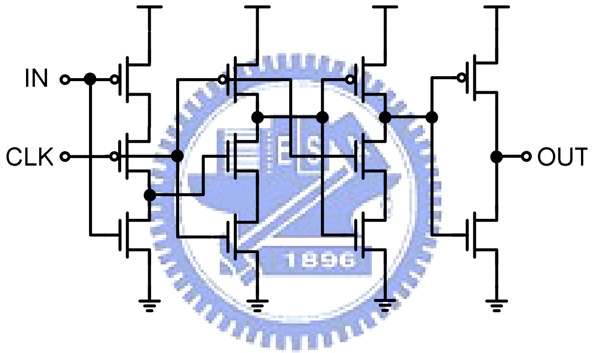
Appendix B

Logic Implementation

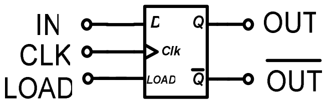
In our proposed CDR, almost all building blocks of circuits are digitally implemented. Thus, many logic gates and D-Flip-Flops must be utilized to make all function correct. In this appendix, we introduce the logic implementation.

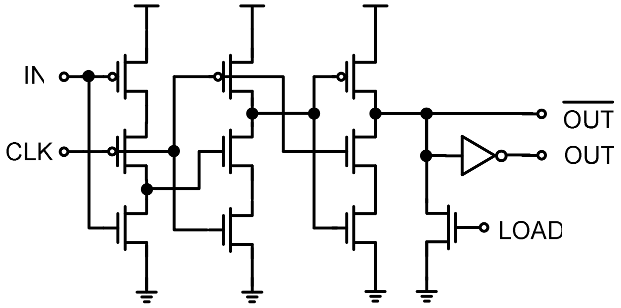
B.1 Conventional TSPC-Type DFF

We describe the conventional implementation for D-Flip-Flop. Our DFF is implemented by True-Single-Phase-Clock-type (TSPC-type) D-Flip-Flop and the circuit implementation is demonstrated in Fig. B-1(a). If our DFF must be given a initial value, we should exploit the DFF with “load” function. Our initial control signal varies from high to low. Thus, we utilize the DFF with LOAD if the initial value is “1” and logic symbol and circuit implementation are illustrated in Fig. B-1(b) and (c), respectively. On the contrary, we utilize the DFF with LOADB if the initial value is “0” and logic symbol and circuit implementation are illustrated in Fig. B-1(d) and (e), respectively. Moreover, logic symbol and circuit implementation of the DFF with “reset” function is described in Fig. B-1(f) and (g), respectively.



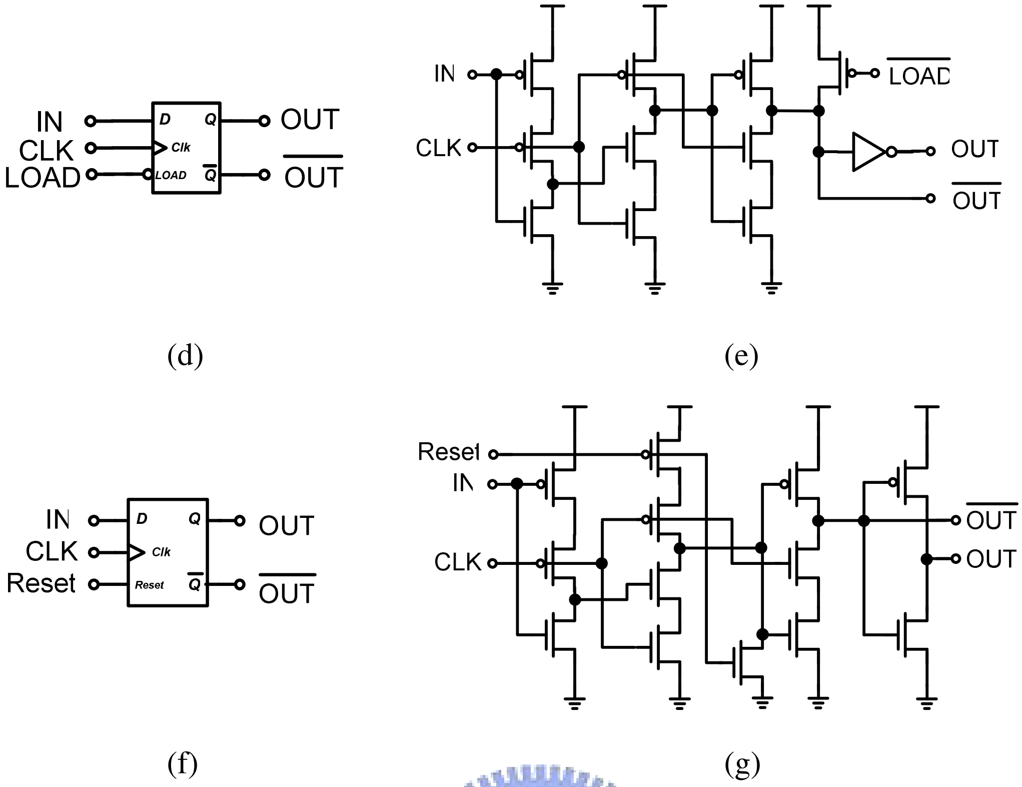
(a)





(b)

(c)



(d) (e)

(f) (g)

Fig. B-1 (a) Conventional DFF Circuit Implementation

(b) DFF with LOAD Logic Symbol

(c) DFF with LOAD Circuit Implementation

(d) DFF with LOADB Logic Symbol

(e) DFF with LOAD Circuit Implementation

(f) DFF with RESET Logic Symbol

(g) DFF with RESET Circuit Implementation

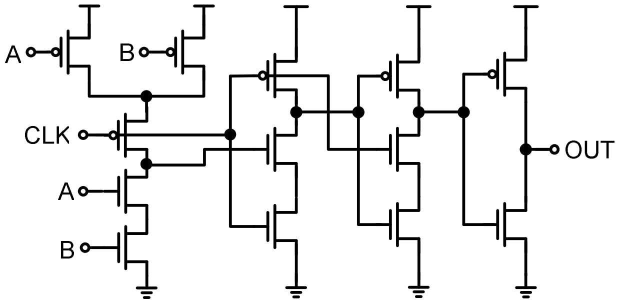
B.2 DFF with Embedded Logic Gate

There are many logic gates in our digital building blocks. However, our CDR operates at high speed so that we have to reduce the gate delay of all digital circuits.

The most critical delay path is the pass where must pass through the D-Flip-Flops. Our DFF is implemented by True-Single-Phase-Clock-type (TSPC-type) DFF. We can eliminate the delay path if we combine the TSPC-type DFF and logic gate which is in the front of the DFF together. In other words, The front-end logic gate is embedded into the TSPC-type DFF. Many kinds of logic gates can be embedded, such as NAND, NOR, XOR, and so on. We describe the detail circuit implementation below.

B.2.1 D-Flip-Flop with Embedded NAND

Fig. B-2(a) shows the circuit implementation of the DFF with embedded NAND. As mentioned before, if our DFF with embedded NAND must be given a initial value, we should exploit the DFF with "load" function. The logic symbol and circuit implementation of the DFF with embedded NAND and LOAD are illustrated in Fig. B-2(b) and (c), respectively. On the contrary, the logic symbol and circuit implementation of the DFF with embedded NAND and LOADB are demonstrated in Fig. B-2(d) and (e), respectively.



(a)

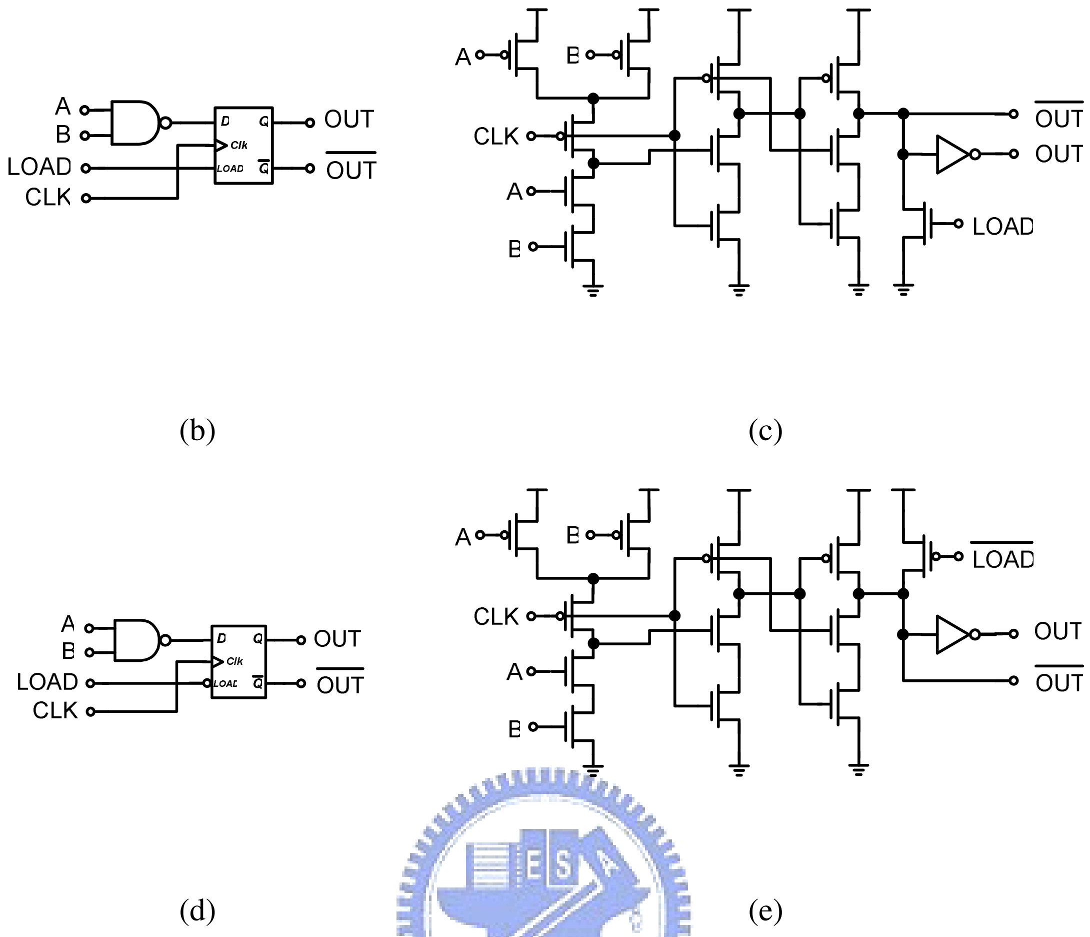


Fig. B-2 (a) DFF with embedded NAND Circuit Implementation

(b) DFF with embedded NAND and LOAD Logic Symbol

(c) DFF with embedded NAND and LOAD Circuit Implementation

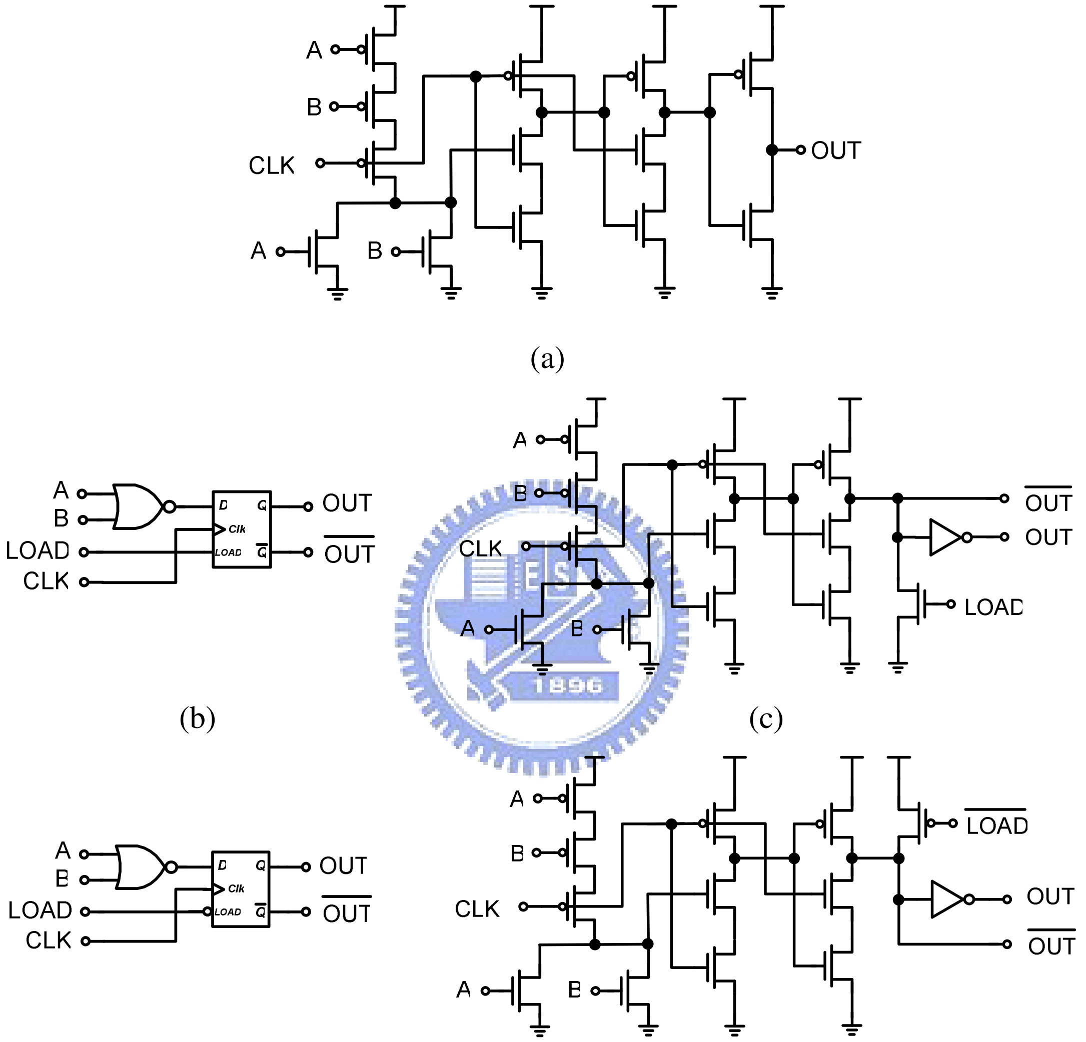
(d) DFF with embedded NAND and LOADB Logic Symbol

(e) DFF with embedded NAND and LOADB Circuit Implementation

B.2.2 D-Flip-Flop with Embedded NOR

Fig. B-3(a) shows the circuit implementation of the DFF with embedded NOR. The logic symbol and circuit implementation of the DFF with embedded NOR and

LOAD are illustrated in Fig. B-3(b) and (c), respectively. On the contrary, the logic symbol and circuit implementation of the DFF with embedded NOR and LOADB are demonstrated in Fig. B-3(d) and (e), respectively.



(a)

(b)

(c)

(d)

(e)

Fig. B-3 (a) DFF with embedded NOR Circuit Implementation

(b) DFF with embedded NOR and LOAD Logic Symbol

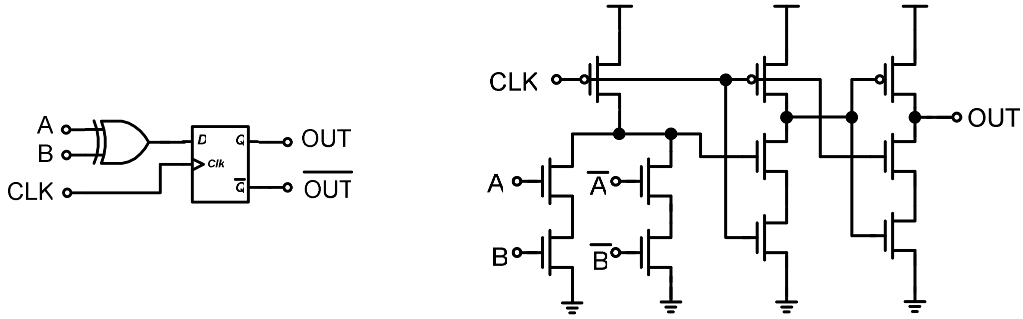
(c) DFF with embedded NOR and LOAD Circuit Implementation

(d) DFF with embedded NOR and LOADB Logic Symbol

(e) DFF with embedded NOR and LOADB Circuit Implementation

B.2.3 D-Flip-Flop with Embedded XOR

Fig. B-4(a) and (b) illustrates the logic symbol and circuit implementation of the DFF with embedded XOR, respectively.



(a) (b)

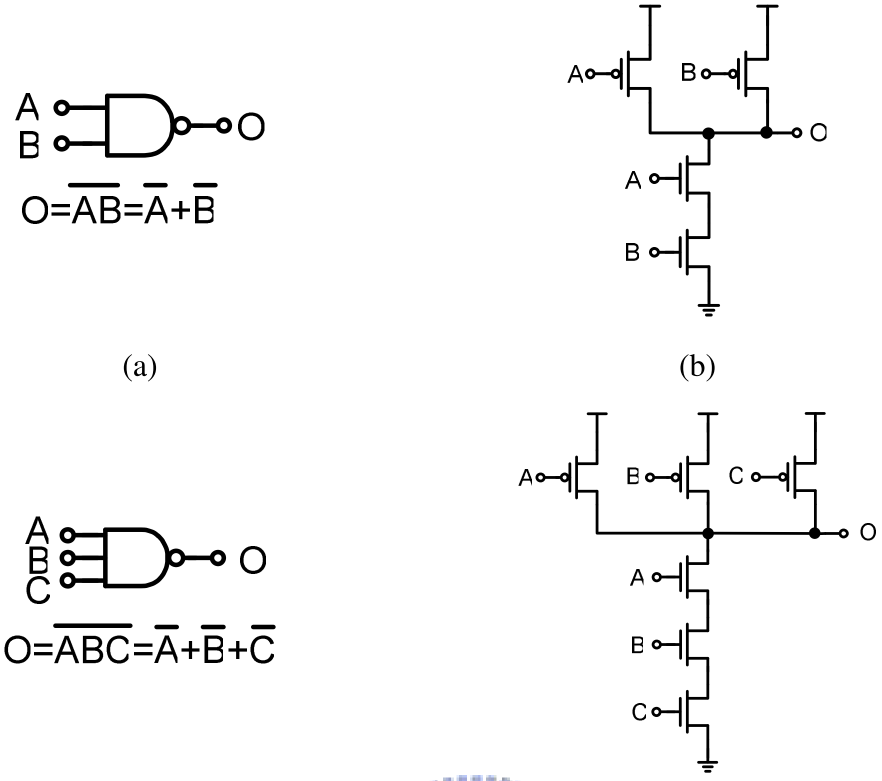
Fig. B-4 (a) DFF with embedded XOR Logic Symbol

(b) DFF with embedded XOR Circuit Implementation

B.3 Basic Logic Gate Implementation

B.3.1 NAND Logic

Fig. B-5(a) and (b) illustrates the logic symbol and circuit implementation of the 2-input NAND gate, respectively, while Fig. B-5(c) and (d) illustrates the logic symbol and circuit implementation of the 3-input NAND gate, respectively. Our NAND logic is all implemented by complementary CMOS type.


A
B
O
$O=\overline{AB}=\overline{A}+\overline{B}$
A
B
O
A
B
A
B
C
O
$O=\overline{ABC}=\overline{A}+\overline{B}+\overline{C}$
A
B
C
O
A
B
C

(a) (b)

(c) (d)

Fig. B-5 (a) 2-Input NAND Gate Logic Symbol

(b) 2-Input NAND Gate Circuit Implementation

(c) 3-Input NAND Gate Logic Symbol

(d) 3-Input NAND Gate Circuit Implementation

B.3.2 NOR Logic

Fig. B-6(a) and (b) demonstrates the logic symbol and circuit implementation of the 2-input NOR gate, respectively, while Fig. B-6(c) and (d) demonstrates the logic symbol and circuit implementation of the 3-input NOR gate, respectively. Our NOR logic is all implemented by complementary CMOS type.

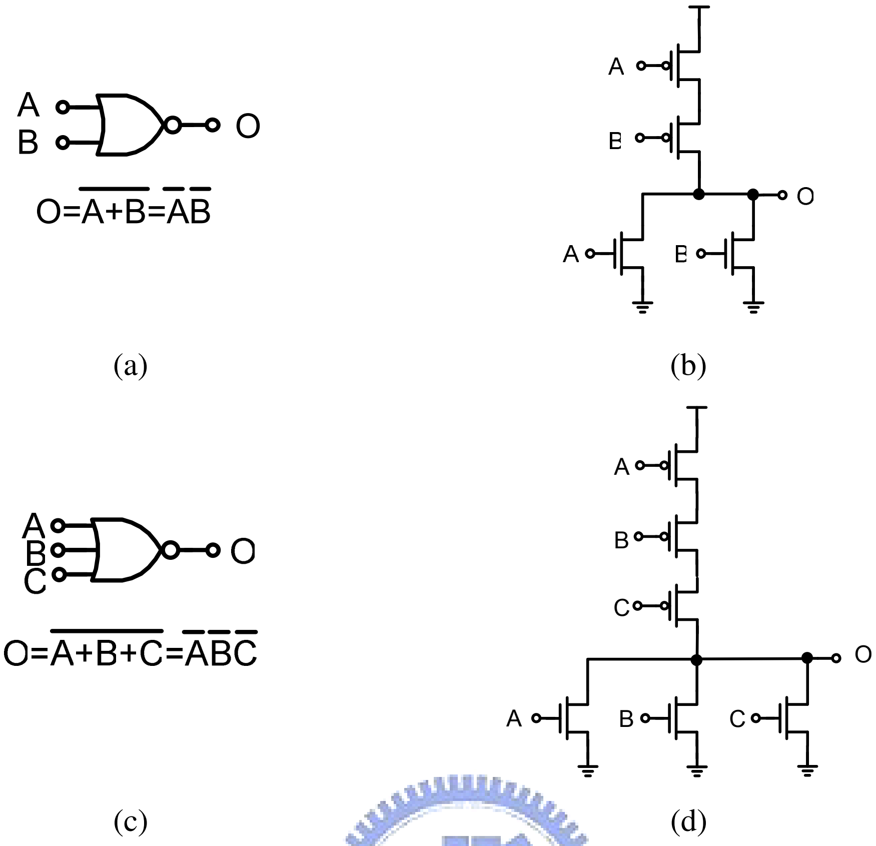


Fig. B-6 (a) 2-Input NOR Gate Logic Symbol

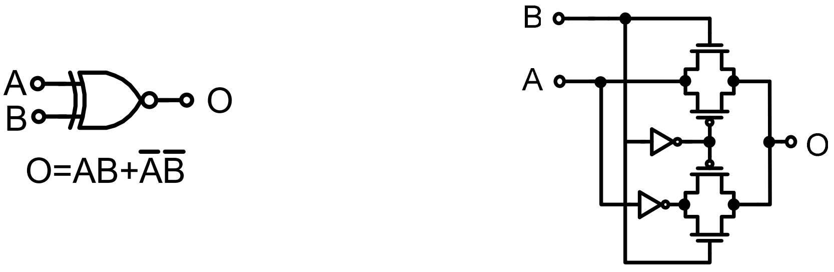
(b) 2-Input NOR Gate Circuit Implementation

(c) 3-Input NOR Gate Logic Symbol

(d) 3-Input NOR Gate Circuit Implementation

B.3.3 XOR Logic

Fig. B-7(a) and (b) demonstrates the logic symbol and circuit implementation of the 2-input XOR gate, respectively. Our XOR logic is all implemented by transmission-gate type.



(a) (b)

Fig. B-7 (a) 2-Input XOR Gate Logic Symbol

(b) 2-Input XOR Gate Circuit Implementation

References

[1] Hitoyuki Tagami, et al. “A Burst-Mode Bit-Synchronization IC With Large Tolerance for Pulse-Width Distortion for Gigabit Ethernet PON,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, pp. 2555–2565, Nov. 2006.

[2] *Ethernet in the First Mile Task Force*, IEEE 802.3ah , IEEE Standards Association.

[3] Jri Lee , and Mingchung Liu , “A 20Gb/s Burst-Mode CDR Circuit Using Injection-Locking Technique ,” *IEEE ISSCC Dig. Tech. Papers*, pp. 46-47, Feb., 2007.

[4] Lan-Chou Cho , Chihun Lee , and Shan-Iuan Liu , “A 33.6-to-33.8Gb/s Burst-Mode CDR in 90nm CMOS ,” *IEEE ISSCC Dig. Tech. Papers*, pp. 48-49, Feb., 2007.

[5] Jri Lee and Behzad Razavi , “A 40Gb/s Clock and Data Recovery Circuit in 0.18μm CMOS Technology,” *IEEE Journal of Solid-State Circuits*, vol. 38 , no. 12 , pp. 2181-2188 , Dec. 2003.

[6] J. Savoj and Behzad Razavi, “A 10 Gb/s CMOS clock and data recovery circuit with a half-rate binary phase/frequency detector,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, p. 13, Jan. 2003.

[7] R.-J. Yang, S.-P. Chen, and S.-I. Liu, “A 3.125 Gb/s clock and data recovery circuit for the 10 Gbase-LX4 Ethernet,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, p. 1356, Aug. 2004.

[8] A. Rezayee and K. Martin, “A 9-16 Gb/s clock and data recovery circuit with three-state phase detector and dual-path loop architecture,” in *Proc. 29th Europe Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2003, pp. 683–686.

[9] E. Nosaka, H. Sano, K. Ishii, M. Ida, K. Kurishima, S. Yamahata, T. Shibata, H. Fukuyama, M. Yoneyama, T. Enoki, and M. Muraguchi, “A 39-to-45 Gb/s multi-data-rate clock and data recovery circuit with a robust lock detector,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 8, pp. 1361–1365, Aug. 2004.

[10] S. Jonathan E. Rogers, and John R. Long, “A 10Gb/s CDR/DEMUX with LC Delay Line VCO in 0.18μm CMOS,” *IEEE ISSCC Dig. Tech. Papers*, pp. 254-255, 2002.

[11] M. Meghelli et al., “SiGe BiCMOS 3.3-V Clock and Data Recovery Circuits for 10-Gb/s Serial Transmission System,” *IEEE Journal of Solid-State Circuits*, vol.35, No. 12, pp. 1992-1995, Dec. 2000.

[12] Y. M. Greshishchev et al., “SiGe Clock and Data Recovery IC with Linear-Type

PLL for 10-Gb/s SONET Application," *IEEE Journal of Solid-State Circuits*, vol.35, No. 9, pp. 1353-1359, September 2000.

[13] S. G. Georgiou, Y. Baeyens et al., "Clock and Data Recovery IC for 40-Gb/s Fiber-Optical Receiver," *IEEE Journal of Solid-State Circuits*, vol.37, No. 9, pp. 1120-1125, September 2002.

[14] Michael H. Perrott, et al , "A 2.5-Gb/s Multi-Rate 0.25-μm CMOS Clock and Data Recovery Circuit Utilizing a Hybrid Analog/Digital Loop Filter and All-Digital Referenceless Frequency Acquisition , "*IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp.2930-2944, Dec. 2006.

[15] N. Ishihara and Y. Akazawa, "A monolithic 156-Mb/s clock and data recovery PLL circuit using the sample-and-hold technique," *IEEE J.Solid-State Circuits*, vol. 29, no. 12, pp. 1566–1571, Dec. 1994.

[16] K. Yamashita, M. Nakata, N. Kamogawa, O. Yumoto, and H. Kodera, "Compact-same-size 52- and 156 Mbit/s SDH optical transceiver modules," *J. Lightw. Technol.*, vol. 12, no. 9, pp. 1607–1615, Sep. 1994.

[17] Chih-Kong Ken Yang, *et al.* , "A 0.5 μ m CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling," in *IEEE J. Solid-State Circuits*, vol. 33, pp. 713-722, May. 1998.

[18] Che-Fu Liang , et al ,"A 2.5Gbps Burst-Mode Clock and Data Recovery Circuit , " in *IEEE ASSCC* , P2-23 , 2005

[19] Alan Li , Julien Faucher , and David V. Plant , "Burst-Mode Clock and Data Recovery in Optical Multi-access Networks Using Broad-Band PLLs , " *IEEE Photonics Technology Letters* , Vol.18 , No.1 , Jan , 2006.

[20] Masafumi Nogawa, *et al*, "A 10 Gb/s burst-mode CDR IC in 0.13 μ m CMOS, " in *ISSCC Digest of Technical Papers*, pp. 228-229 , Feb. 2005.

[21] Toru Iwata,Takashi Hirata,Hirokazu Sugimoto,Hiroshi Kimura,and Takefumi Yoshikawa,"A 5Gbps CMOS Frequency Tolerant Multi Phase Clock Recovery Circuit,"*IEEE Symposium On VLSI Circuits Dig. Tech. Papers* , 2002 .

[22] Y. Ota, R. G. Swartz, V. D. Archer, III, K. Korotoky, M. Banu, and A. E. Dunlop, "High-speed, burst-mode packet-capable optical receiver and instantaneous clock recovery for optical bus operation," *J. Lightw. Technol.*, vol. 12, no. 2, pp. 325–330, Feb. 1994.

[23] M. Banu and A. E. Dunlop, "Clock recovery circuits with instantaneous locking," *Electron. Lett.*, vol. 28, no. 23, pp. 2127–2130, Nov. 1992.

[24] M. Nakamura, N. Ishihara, and Y. Akazawa, "A 156 Mbps CMOS clock recovery circuit for burst-mode transmission," in *Symp. VLSI Circuits Dig. Tech. Papers*, Dec. 1996, pp. 122–123.

[25] Y. Yamada, S. Mino, and K. Habara, "Ultra-fast clock recovery for burst-mode

optical packet communication," in *Proc. OFC 1999*, Feb. 1999, pp. 114–116.

[26] Hitoyuki Tagami, "A Burst-Mode Bit-Synchronization IC With Large Tolerance for Pulse-Width Distortion for Gigabit Ethernet PON," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2555–2565, Nov. 2006.

[27] S. Kobayashi and M. Hashimoto, "A multibitrate burst-mode CDR circuit with bit-rate discrimination function from 52 to 1244 Mb/s," *IEEE Photon. Technol. Lett.*, vol. 13, no. 11, pp. 1221–1223, Nov. 2001.

[28] S. Kimura, M. Nogawa, K. Nishimura, T. Yoshida, K. Kumozaki, S. Nishihara, and Y. Ohtomo, "A 10-Gbit/s CMOS-burst-mode clock and data recovery IC for a WDM/TDM-PON access network," in *Proc. 17th Annu. Meeting IEEE Lasers and Electro-Optics Society (LEOS 2004)*, Jul. 2004, pp. 310–311.

[29]Wei-Zen Chen and Chin-Yuan Wei, "High-speed clock and data recovery circuit," *U.S. patent pending. and R.O.C. patent granted .*

[30] Wei-Zen Chen , Chin-Yuan Wei , and Jen-Wen Chen , "A wide range fast lock burst-mode clock and data recovery circuit," in *IEEE ASSCC* , 2006

[31] J. Alexander, "Clock recovery from random binary signals," *Electron. Lett.*, vol. 11, no. 22, pp. 541–542, Oct. 1975.

[32] C. R. Hogge Jr., "A self correcting clock recovery circuit," *IEEE Trans. Electron Devices*, vol. ED-32, no. 12, pp. 2704–2706, Dec. 1985.

[33] M. Fukaishi, K. Nakamura, H. Heiuchi, Y. Hirota, Y. Nakazawa, H. Ikeno, H. Hayama, and M. Yotsuyanagi, "A 20-Gb/s CMOS multichannel transmitter and receiver chip set for ultra-high-resolution digital displays," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1611–1618, Nov. 2000.

[34] S. Sidiropoulos and M. A. Horowitz, "A semidigital dual delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1683–1692, Nov. 1997.

[35] R. Kreienkamp, U. Langmann, C. Zimmermann, T. Aoyama, and H. Siedhoff, "A 10Gb/s CMOS clock and data recovery circuit with an analog phase interpolator," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 736–743, Mar. 2005.

[36] Christian 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.

[37] R. C.Walker, "Designing bang-bang PLLs for clock and data recovery in serial data transmission systems," in *Phase-Locking in High-Performance Systems — From Devices to Architectures*, B. Razavi, Ed.. New York: IEEE Press, 2003, pp. 34–45.

[38] C. Kromer, "10 Gb/s to 40 Gb/s Receiver for High-Density Optical Interconnects in 80-nm CMOS," Konstanz, Germany: Hartung-Gorre, 2006 .

[39] Tsung-Hsien Lin *et al*, “A 900-MHz 2.5-mA CMOS Frequency Synthesizer with an Automatic SC Tuning Loop,” *IEEE J. Solid-State Circuits*, vol. 36, no. 3, pp.424-431 , March 2001

[40] Tsung-Hsien Lin , and Yu-Jen Lai, “An Agile VCO Frequency Calibration Technique for a 10-GHz CMOS PLL , ” *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp.340-349 , Feb. 2007

[41] A. Aktas , and M. Ismail , “CMOS PLL calibration techniques, ” *IEEE Circuits Devices Magazine*, vol. 20, no. 5, pp. 6-11, Sep./Oct. 2004 .

[42] W. B. Wilson, *et al*, “A CMOS self-calibrating frequency synthesizer, ” *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp.1437-1444 , Oct 2000 .

[43] C.F Liang , et al. “A 2.5Gbps Burst-Mode Clock and Data Recovery Circuit, ” *IEEE ASSCC*, P2-23 , 2005.

[44] R-J Yang ,et al. “A 155.52Mbps-3.125Gbps Continuous-Rate Clock and Data Recovery Circuit ,” *IEEE JSSC*, Vol.41, No.6, AUGUST 2006.

簡 歷

姓名 ： 徐 國 慶 Kuo-Ching Hsu

性別 ： 男

出生日 ： 1982.01.18

住址 ： 台北市信義區莊敬路311號6樓

學歷 ：	台北市立吳興國民小學	(77年9月 ~ 83年6月)
	台北市立金華國民中學	(83年9月 ~ 86年6月)
	台北市立建國高級中學	(86年9月 ~ 89年6月)
	國立交通大學電信工程系	(89年9月 ~ 93年6月)
	國立交通大學電子研究所碩士班	(93年9月 ~ 96年4月)

論文名稱 ： 一個操作在1.25到6 Gbps脈衝式時脈資料回復電路

A 1.25~6 Gbps Burst-Mode Clock and Data Recovery Circuit

修習課程 ：

高等數位訊號處理	IEE7732	劉志尉教授	86
射頻積體電路	IEE5646	郭建男教授	92
微波工程(I)	ECM5501	郭仁財教授	86
展頻通訊	ECM5105	黃家齊教授	88
類比積體電路(I)	IEE6703	吳介琮教授	98
類比積體電路(II)	IEE6736	吳介琮教授	90
數位積體電路	IEE7721	柯明道教授	92
有線傳輸通信積體電路設計	IEE5692	陳巍仁教授	90
積體電路之靜電防護設計	IEE5639	柯明道教授	92
半導體物理及元件(I)	IEE5561	雷添福教授	94

著作 ：

[1]國際SCI期刊(Journal)論文:

Wei-Zen Chen , Wen-Hui Chen and Kuo-Ching Hsu , “ 3-Dimensional Fully Symmetric Inductors, Transformer, and Balun in CMOS Technology “ , accepted by *IEEE Transaction on Circuits and Systems I : Regular Papers* .

[2]國際研討會(Conference)論文:

Wei-Zen Chen and Kuo-Ching Hsu , “ Miniaturized 3-Dimensional Transformer Design ,” *IEEE Custom Integrated Circuit Conference (CICC)* , pp. 285-288 , September , 2005 .

[3]“快速鎖定時脈與資料回復電路設計”，九十五年度大學院校混合訊號與射頻電路設計(MSR)碩士論文觀摩競賽佳作獎