Analysis and Design of a Phase Interpolator for Clock and Data Recovery

Sun Yehui[1,†] and Jiang Lixin[2]

(1 *The College of Information Science and Technology, Nankai University, Tianjin 300071, China*)
(2 *IDT (Shanghai) Co., Ltd, Shanghai 200233, China*)

Abstract: In this paper, a detailed analysis of a phase interpolator for clock recovery is presented. A mathematical model is setup for the phase interpolator and we perform a precise analysis using this model. The result shows that the output amplitude and linearity of phase interpolator is primarily related to the difference between the two input phases. A new encoding pattern is given to solve this problem. Analysis in the circuit domain was also undertaken. The simulation results show that the relation between RC time-constant and time difference of input clocks affects the linearity of the phase interpolator. To alleviate this undesired effect, two adjustable-RC buffers are added at the input of the PI. Finally, a 90nm CMOS phase interpolator, which can work in the frequency from 1GHz to 5GHz, is proposed. The power dissipation of the phase interpolator is 1mW with a 1.2V power supply. Experiment results show that the phase interpolator has a monotone output phase and good linearity.

Key words: phase interpolator; clock and data recovery; CMOS
EEACC: 1220; 1280
CLC number: TN43 **Document code**: A **Article ID**: 0253-4177(2008)05-0930-06

1 Introduction

In the widely used serial communication, the data will be transmitted from transmitter to the receiver without a synchronous clock. The received data suffer from asynchronous and noise effects[1]. To recover the data, the system needs to extract a clock and use it to synchronize and "clear" the data. This system is called clock and data recovery (CDR). However, the received data accumulates jitter and noise during transmission. To recover the data correctly and decrease the bit error rate (BER), the extracted clock needs to track the phase of received data timely and accurately. Thus, adjusting the phase of the recovery clock based on the received data is a chief function in a CDR system.

There are several methods to adjust the phase of the recovery clock. Traditional CDRs based on the phase lock loop (PLL) change the phase of the recovery clock through changing the instant clock frequency, as shown in Fig. 1(a)[2]. The phase detector (PD) detects the difference between received data and the recovered clock, and this error information is low-pass filtered and sent to the voltage controlled oscillator (VCO). The VCO changes the phase of output clock by changing its instant frequency. After convergence, the phase error between the data and clock is eliminated. PLL-based CDRs are easy to implement because of their classical architecture. But they consume large power and cause crosstalk between the different channels because they require an individual VCO with each channel. Another kind of

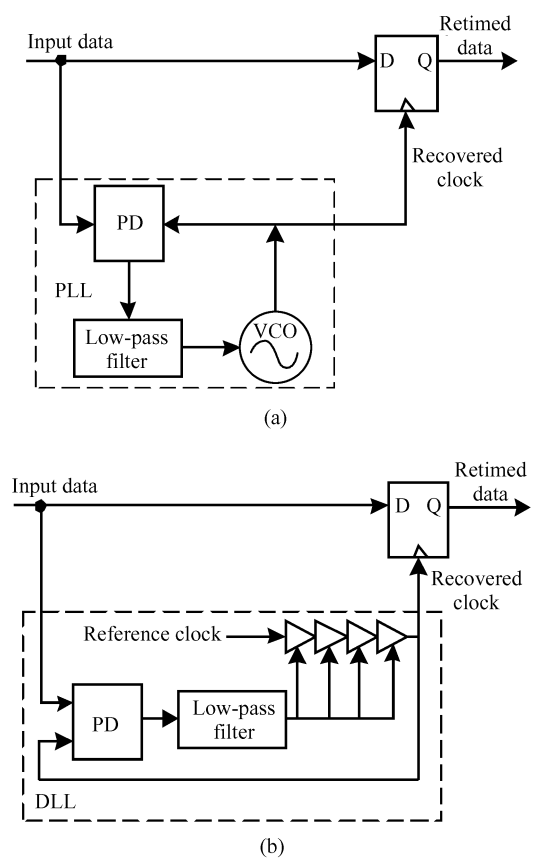

Fig. 1 Clock and data recovery circuit (a) Based on PLL; (b) Based on DLL

† Corresponding author. Email: yehui.sun@yahoo.com.cn
Received 23 October 2007, revised manuscript received 10 December 2007

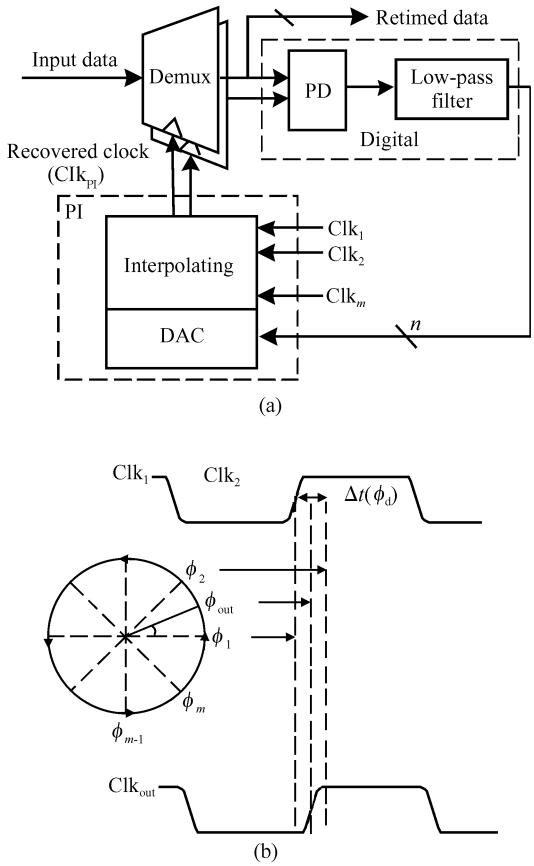

Fig. 2　Clock recovery circuit based on phase interpolator　(a) Architecture; (b) Phase interpolating

delay lock loop (DLL) based CDR is also widely used[3]. The architecture is shown in Fig. 1 (b). Unlike PLL-based CDRs, DLL-based CDRs do not change the frequency of the output clock. Instead, they adjust the phase of the recovery clock by tuning the delay time of the delay cell. DLL-based CDRs can share one VCO block with multi-channels to avoid the crosstalk.

The PLL- and DLL-based CDRs are sensitive to process because they are usually implemented with analog circuits. A phase interpolator (PI)-based CDR is widely used in the serial communication design[4]. As shown in Fig. 2, unlike the previous two architectures, the inputs of CDR are a set of different phase clocks ($\mathrm{Clk}_1 \sim \mathrm{Clk}_m$) from the VCO. $\phi_1 \sim \phi_m$ are the phases of $\mathrm{Clk}_1 \sim \mathrm{Clk}_m$ and n is the digital control word. PI adjusts the weights of the adjacent input clocks through a digital to analog converter (DAC) to generate an output clock with any phase between them. Then, the output clock is used to sample the received data. ϕ_{out} is the phase of the PI output clock ($\mathrm{Clk}_{\mathrm{PI}}$). This architecture shares a common VCO to avoid the crosstalk and it usually uses digital control instead of analog to release the process variation.

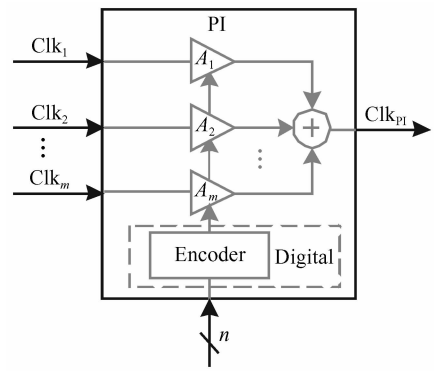

Fig. 3　Mathematic model of PI

2　Phase interpolator modeling

The two most desirable properties for a phase interpolator are:

(1) Having a monotonic transfer characteristic.

(2) Having a linear transfer characteristic.

These two properties are important to the PI and CDR. Non-ideal effects in the system and circuit will degrade the monotony and linearity of the PI and cause errors in the CDR. This section analyzes the non-ideal effect from the a mathematic view. The mathematic representation of PI in Fig. 2 is shown in Fig. 3. In theory, the output phase of PI ($\mathrm{Clk}_{\mathrm{PI}}$) is proportional with the input control bits. This is given by

$$\phi_{\mathrm{out}} = k_{\mathrm{PI}} n, \quad 0 \leqslant n \leqslant N \tag{1}$$

where k_{PI} is the gain of the PI and n is the input control word where $n = 0$ to N. Equation (1) shows the phase of $\mathrm{Clk}_{\mathrm{PI}}$ (ϕ_{out}) increases from 0 to 2π as n increases from 0 to N. Equation (1) demonstrates the transfer curve between ϕ_{out} and n will be monotonic and linear as long as k_{PI} remains constant. Practically, ϕ_{out} is derived from interpolating the phase of input clocks.

As a rule, $\phi_1 \sim \phi_m$ separate from 0 to 2π equally. So, we only need to analyze the interpolating mechanism between Clk_1 and Clk_2. We approximate $\mathrm{Clk}_1 = \sin(\omega t)$ and $\mathrm{Clk}_2 = \sin(\omega t + \phi_{\mathrm{d}})$ where ω is circular frequency of input clock and $\phi_{\mathrm{d}} = \phi_2 - \phi_1$ with the assumption of $\phi_1 = 0$. All weight coefficients are zero except A_1 and A_2. The output of $\mathrm{Clk}_{\mathrm{PI}}$ can be calculated

$$\mathrm{Clk}_{\mathrm{PI}} = A_1 \sin(\omega t) + A_2 \sin(\omega t + \phi_{\mathrm{d}}) \tag{2}$$

According to the theorem of trigonometric functions, we obtain

$$\begin{cases} \sin(\omega t + \phi_{\mathrm{d}}) = \sin(\omega t)\cos\phi_{\mathrm{d}} + \cos(\omega t)\sin\phi_{\mathrm{d}} \\ p = A_1 + A_2\cos\phi_{\mathrm{d}} \\ q = A_2\sin\phi_{\mathrm{d}} \end{cases} \tag{3}$$

Equation (2) can be rewritten as

$$\mathrm{Clk}_{\mathrm{PI}} = \sqrt{p^2 + q^2}\sin(\omega t + \phi_{\mathrm{out}}) \tag{4}$$

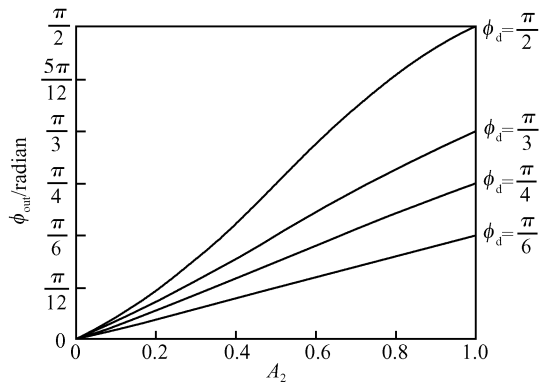

Fig. 4 PI transfer function with varying ϕ_d

Equation (4) demonstrates the phase and magnitude of output clock are both variable due to the interpolating. The phase and magnitude are determined by A_1, A_2, and ϕ_d

$$\sqrt{p^2 + q^2} = \sqrt{A_1^2 + 2A_1A_2\cos\phi_d + A_2^2} \quad (5)$$

$$\phi_{out} = \arctan\left(\frac{A_2\sin\phi_d}{A_1 + A_2\cos\phi_d}\right) \quad (6)$$

A nonidentity can be found by comparing Eqs. (1) and (6), and the actual output phase is a function of A_1, A_2, and ϕ_d. A_1 and A_2 are encoded from the control word n. Generally, they meet the normalized Eq. (7) for the simple linear encoding implementation.

$$A_1 + A_2 = 1 \quad (7)$$

As shown in the Fig. 4, ϕ_{out} increases from 0 to ϕ_d while A_2 increases from 0 to 1. The linearity of ϕ_{out} is degraded as ϕ_d increases. The linearity is hardly damaged when $\phi_d > \pi/2$ because there is a mismatch between the ideal Eq. (1) and the nonlinearity Eq. (6). Two methods can be used to reduce this undesired effect. First, choosing a smaller ϕ_d. Figure 4 shows that the linearity will be good enough in the case of $\phi_d < \pi/3$. Intuitively, this is because the nonlinearity will be compensated more as the linear interpolating point increases. However, it will require more input clock phases and make clock generation difficult. The second method is choosing a nonlinear encoding pattern to compensate the nonlinearity of Eq. (6), which releases the clock generation circuits. Usually the encoder is implemented in a digital domain using a look-up table. It will not occupy much more die area than the linear encoding pattern. For example, when choosing $\phi_d = \pi/2$, equations (5) and (6) can be rewritten as

$$\sqrt{p^2 + q^2} = \sqrt{A_1^1 + A_2^2} \quad (8)$$

$$\phi_{out} = \arctan\left(\frac{A_2}{A_1}\right) \quad (9)$$

Comparing Eq. (1) and Eq. (9), we find ϕ_{out} is not a linear function of n but an inverse trigonometric

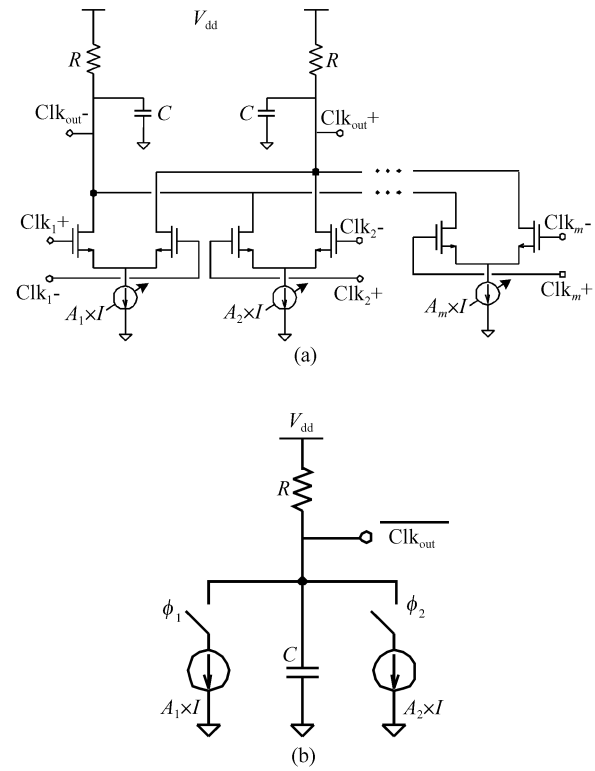

Fig. 5 PI with delay cells (a) Circuits; (b) Simplified model

function of A_2/A_1. We choose a new encoding pattern as in Eqs. (10) and (11) instead of Eq. (7).

$$A_1^2 + A_2^2 = 1 \quad (10)$$

$$\frac{A_2}{A_1} = \tan(k_{PI}n) \quad (11)$$

We obtain a linear phase and constant magnitude output clock. The phase and magnitude of the PI output clock will be

$$\sqrt{p^2 + q^2} = 1 \quad (12)$$

$$\phi_{out} = k_{PI}n \quad (13)$$

Figure 4 indicates that the INL will be larger than ± 4 degree when using a linear encoding pattern with $\phi_d = \pi/2$. Except for the nonideal factor during circuit implementation, the INL will be cancelled completely when using Eq. (11) as the encoding pattern.

3 Circuit implementation

The phase interpolator is most critical circuit in the design of the clock and data recovery. As shown in Fig. 2(a), the PI is an m-inputs buffer that receives m clocks, Clk_1 to Clk_m, and generates the output clock. The m-inputs buffer consists of m identical delay cells whose outputs are shorted as shown in Fig. 5 (a). The current sources of the delay cells compose a current mode DAC. The control signals of DAC come from the encoder output. The phase and magnitude of the output clock are determined by the control sig-

nals. The encoder usually is realized in the digital domain and we will not discuss the implementation for simplicity. Ideally, the phase of the output clock is the weighted sum of the phases of input clocks, as discussed in section 2. In practice, when the speed decreases, the input clock resembles a pulse more than a sine wave and the delay cell has a finite output impedance R and capacitance loading C, which we did not mention in section 2. This situation brings more nonideal factors that will degraded the linearity of the PI and should be considered in the circuit design.

As discussed in section 2, the linearity and magnitude of the output clock is affected by the timing difference $\left(\Delta t = \frac{\phi_d}{2\pi} T_{clk}\right)$ between two input phases from a mathematical view. We will illustrate more mechanisms between the output clock and Δt from the circuit design view based on the simplified model shown in Fig. 5(b). R is the output impedance and C is capacitance loading.

We consider how Δt affects the output linearity of PI, choosing Eq. (7) as the encoding rule in analysis for simplicity. As shown in Fig. 5(b), interpolation is accomplished by switching the outputs of the two current sources. The output signal, $\overline{\text{Clk}_{\text{out}}}$ is high when both inputs are low and low when both inputs are high. However, when one input is high and the other is low, $\overline{\text{Clk}_{\text{out}}}$ becomes approximately $V_{dd} - IR/2$ assuming $A_1 = A_2 = 0.5$. This degrades the linearity of PI enormously. The half transition region shown in Fig. 6(a) shrinks as Δt is reduced and eventually disappears, resulting in a smooth transition. For PI to work properly, Δt must be small, namely comparable to one gate delay, so that the transitions overlap to some extent[5].

To verify the above conclusion, an output voltage expression should be derived for analyzing the PI. From the model, consider the input clocks are a pulse, the interpolator output voltage is a function of Δt, and t is given by

$$V_{out}(t) = V_{dd} + RI[A_1 u(t)(e^{-\frac{t}{RC}} - 1) + A_2 u(t - \Delta t)(e^{-\frac{t-\Delta t}{RC}} - 1)] \quad (14)$$

where Δt is the time delay between the two input clocks. Equation (14) shows that the interpolator delay depends not only on the interpolation weight but also on the time delay between the interpolator inputs. Using Eq. (14), the interpolator transfer function can be derived. Figure 6(a) illustrates the transfer function, for varying values of Δt. Figure 6(b) shows that the interpolator transfer function becomes increasingly nonlinear as the delay between the two step inputs becomes larger than the RC time-constant of the circuit. Although the nonlinearity can be alleviated through decreasing Δt, the increased clock phase number would make the implementation more difficult. As mentioned at the beginning of this section, the sharp edge of the input clock causes the difficulties presented in Eq. (14). As shown in Fig. 7, two additional buffers can be added at the input of the PI core. The clock edge can be smoothed by increasing the RC time - constant of the buffer. This method not only increases the interpolator linearity, but it also ensures that the interpolator output does not settle to a value equal to half the final swing, thus increasing

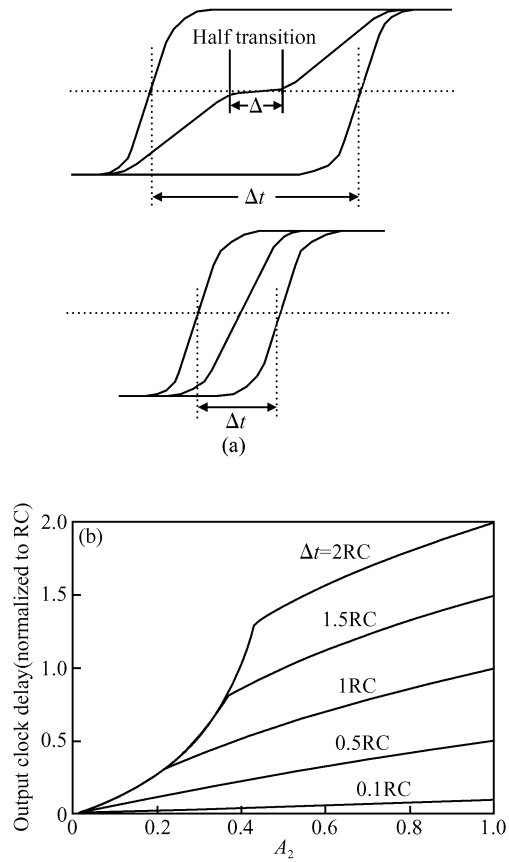

Fig. 6　The relation between Δt and RC time-constant　(a) Half transition; (b) PI transfer function with varying Δt

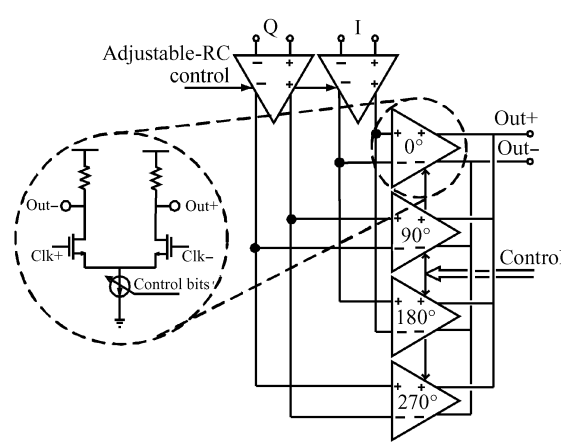

Fig. 7　Phase interpolator with adjustable-RC input buffer

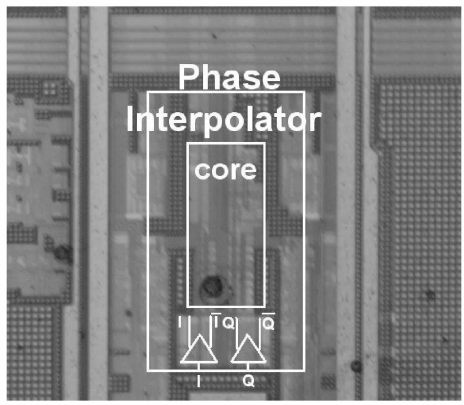

Fig. 8 Microphotograph of PI

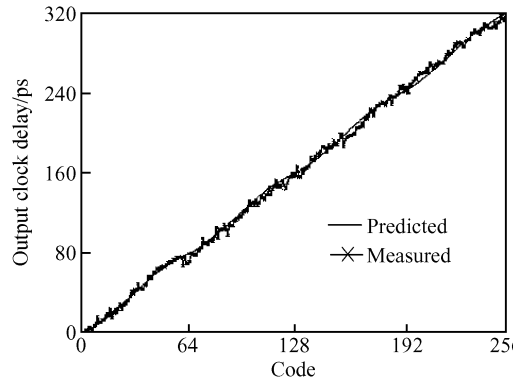

Fig. 9 Measured transfer function of PI

the jitter sensitivity.

The PI we designed has four input clocks ($I = 0$, $Q = \frac{\pi}{2}$, $\overline{I} = \pi$, $\overline{Q} = \frac{3\pi}{2}$) that are generated by two input differential buffers. It has four differential buffers to interpolate the four input clocks, like the architecture shown in Fig. 7. To eliminate the demerit of linear encoding pattern, we use Eqs. (12) and (13) as the encoding rule to mitigate the nonlinearity of PI.

In addition, the nonlinearity of DAC is also a significant problem. References [6～8] have discussed it and we will not research it further in this paper.

4 Experimental results

To verify the analysis above, a phase interpolator was fabricated in a 90nm CMOS process. It consumes 1mW at 1.2V power supply. Figure 8 shows the chip microphotograph.

The input clock frequency range is 1GHz to 5GHz. The transfer function test was done with the 3.125GHz input clocks. Figure 9 shows the transfer function of the PI.

The solid line in Fig. 9 is the curve predicted from the analysis above. The solid line with decussation is the result measured with an Angilent 54854A. Except for the glitch due to the noise from the PCB and the power supply, the measured transfer curve of the PI is monotonic. Measurement was repeated to cancel the random noise and obtain the $|\mathrm{INL}| < 4$ degree, which shows that PI has good linearity.

5 Conclusion

A detail analysis of a phase interpolator for clock recovery is presented. First, a mathematical model is setup for the phase interpolator in section 2. The conclusion shows that the output amplitude and the linearity of the phase interpolator is mainly related to the difference between the two input phases. The second conclusion shows that the RC time-constant also affects the linearity of the PI. Finally, a 90nm CMOS phase interpolator is proposed. The design uses a novel encoding pattern and adjustable-RC buffers to compensate the nonlinearity of the PI. The power dissipation of the PI is 1mW with a 1.2V power supply. Its working range is 1GHz to 5GHz. Testing results show that the phase interpolator has a monotone output phase and good linearity.

References

[1] Razavi B. Prospects of CMOS technology for high-speed optical communication circuits. IEEE J Solid-State Circuits, 2002, 37: 1135

[2] Reinhold M, Dorschky C, Rose E, et al. A fully integrated 40-Gb/s clock and data recovery IC with 1:4 DEMUX in SiGe technology. IEEE J Solid-State Circuits, 2001, 36: 1937

[3] Coban A L, Koroglu M H, Ahmed K A. A 2.5-3.125-Gb/s quad transceiver with second-order analog DLL-based CDRs. IEEE J Solid-State Circuits, 2005, 40: 1940

[4] He M Y, Poulton J. A CMOS mixed-signal clock and data recovery circuit for OIF CEI-6G + backplane transceiver. IEEE J Solid-State Circuits, 2006, 41: 597

[5] Mukhtar S. Interval modulation: a new paradigm for the design of high speed communication systems. PhD Thesis, California Institute of Technology, Pasadena, California, 2004

[6] Johns D A, Martin K. Analog integrated circuit design. New York: John Wiley & Sons, 1997: 463

[7] Groeneveld D W J, Schouwenaars H J, Termeer H A H, et al. A self-calibration technique for monolithic high-resolution D/A converters. IEEE J Solid-State Circuits, 1989, 24: 1517

[8] Van der Plas G A M, Vandenbussche J, Sansen W, et al. A 14-bit intrinsic accuracy Q2 random walk CMOS DAC. IEEE J Solid-State Circuits, 1999, 34: 1708

时钟数据恢复电路中相位插值器的分析和设计

孙烨辉[1,†]　江立新[2]

（1 南开大学信息技术科学学院，天津　300071）
（2 IDT 科技（上海）有限公司，上海　200233）

摘要：分析了应用于时钟恢复电路中的相位插值器.为相位插值器建立了数学模型并基于模型对相位插值器在数学域进行了详细的分析.分析结果表明相位插值器输出时钟的相位和幅度强烈地依赖于插值器输入时钟间的相位差，同时提出一种新的编码方法来补偿相位的非线性.考虑到实际电路中寄生效应，文章同样在电路域中对相位插值器进行了详细分析.通过建立电路模型得到 RC 时间常数和输入时钟间的相差的关系，得到了它对相位插值器线性的影响.在设计中通过在 PI 的输入增加可控 RC 的输入缓冲器来调整输入时钟沿的快慢，从而降低了这种影响.最后利用分析得到的结论，使用 90nm CMOS 工艺设计并制造了一个相位插值器.它的供电电压为 1.2V，功耗为 1mW，工作范围从 1GHz 到 5GHz.测试结果表明，输出相位单调并具有良好的线性度，验证了分析的正确性.

关键词：相位插值器；时钟数据恢复；CMOS
EEACC：1220；1280
中图分类号：TN43　　**文献标识码**：A　　**文章编号**：0253-4177(2008)05-0930-06

† 通信作者.Email：yehui.sun@yahoo.com.cn
2007-10-23 收到，2007-12-10 定稿