

doi:10.13756/j.gtxyj.2020.03.005

光通信系统与网络技术

高速光网络时钟恢复算法设计与实现

徐源浩¹, 吕建新^{2,3}

(1. 武汉邮电科学研究院, 武汉 430074; 2. 光纤通信技术和网络国家重点实验室, 武汉 430074;
3. 烽火通信科技股份有限公司, 武汉 430205)

摘要:针对利用灵活光网络(FlexO)技术实现超 100 Gbit/s 业务传输过程中存在的时钟数据恢复问题,文章基于锁相环频率合成器,设计实现了一种新型的超 100 Gbit/s 光转换单元(OTU)帧结构 OTUC_n到 n 个 100 Gbit/s 帧结构 OTU4 业务数据转变的时钟数据恢复方案,可以在 OTU4 业务频点的一定频偏内,迅速恢复客户时钟,并实现对 OTU4 业务数据的重定时。仿真和实测表明,该方案能够有效满足 FlexO 中的时钟恢复需求,实时恢复的客户侧时钟抖动均 $<1 \times 10^{-6}$,并能确保多路重定时的 OTU4 业务数据持续稳定的向下一级传输。

关键词:灵活光网络;锁相环频率合成器;时钟数据恢复

中图分类号: TN919.3 **文献标志码:** A **文章编号:** 1005-8788(2020)03-0023-07

Design and Implementation of Clock Recovery Algorithm for High Speed Optical Transmission System

XU Yuan-hao¹, Lü Jian-xin^{2,3}

(1. Wuhan Research Institute of Posts and Telecommunications, Wuhan 430074, China;
2. State Key Laboratory of Optical Communication Technologies and Networks, Wuhan 430074, China;
3. FiberHome Telecommunication Technologies Co., Ltd., Wuhan 430205, China)

Abstract: Aiming at the problem of clock data recovery during the transportation of beyond 100 Gbit/s services using Flexible Optical Network (FlexO) technology, this paper designs and implements a new clock data recovery solution based on phase-locked loop frequency synthesizer to transform OTUC_n to n OTU4 service data. Within a certain frequency offset of the OTU4 service frequency point, the customer clock is quickly recovered, and the retiming of the OTU4 service data is realized. Simulation and actual measurement prove that this solution can effectively meet the clock recovery requirements in FlexO. The client-side clock jitter of real-time recovery is less than 1×10^{-6} , and it can ensure the continuous and stable transportation of OTU4 business data of multiple channels to the next stage.

Key words: FlexO; phase-locked loop frequency synthesizer; clock and data recovery

0 引 言

目前,100 Gbit/s 的光网络技术已经发展得十分成熟,相应的 100 Gbit/s 设备已在全球范围内大规模商用,然而伴随着云计算、虚拟现实(Virtual Reality, VR)和超高清视频等业务不断发展,对光传输网提出了超 100 Gbit/s 的业务传输需求。灵活光网络(Flexible Optical Network, FlexO)作为一种信号互联的接口技术,提出了超 100 Gbit/s 业务的实施传输方案,通过绑定多个标准速率接口来实现超 100 Gbit/s 业务的传输,然而在其方案的多个阶段,仍存在较多的技术问题。从线路侧超 100 Gbit/s 的业务数据恢复出 100 Gbit/s 业务数据的过程中,需要建立相应的时钟数据恢复(Clock and Data Recovery, CDR)模块从业务码流中恢复客户时钟,并通过相应的接口将恢复出来的

100 Gbit/s 的业务数据送入客户侧光模块。FlexO 复杂的级联结构以及每一级不同速率的映射与解映射行为,都会不可控地导致客户侧实际数据吞吐速率发生偏移。这就需要 CDR 模块首先能够感知客户侧的频偏信息,并在对应客户侧频偏状况下迅速完成客户时钟的恢复与数据的重定时功能,从而确保高速光传输网能够长期稳定地工作。本文通过建立 CDR 算法输出频偏量,调控锁相环频率合成器生成所需时钟的方式,设计实现了一种新型的全数字 CDR 电路,在 100 Gbit/s 业务的一定频偏范围内,均可以迅速恢复客户时钟,确保 100 Gbit/s 业务长期有效地传输。

1 FlexO 中的 CDR 需求

在光传输网(Optical Transport Network, OTN)中,光转换单元(Optical Transform Unit,

收稿日期:2019-10-14

作者简介:徐源浩(1994—),男,河南三门峡人。硕士,主要研究方向为光网络、算法设计以及 FPGA 逻辑设计。

通信作者:徐源浩,硕士。E-mail:lessbean_94@sina.cn

OTU) 拥有多级速率数据传输的帧结构, 其中 OTU4 和 OTUC_n 分别为 100 Gbit/s 和超 100($n \times 100$) Gbit/s 的光通路帧结构。FlexO 技术主要通过复用多路 100 Gbit/s 光模块来实现超 100 Gbit/s 的 OTUC_n 业务传输。在发送端, FlexO 首先将 OTUC_n 业务信号拆分为 n 个 OTU4 信号, 然后将 OTU4 信号映射为 100 Gbit/s 的 FlexO 帧结构信

号, 再将其绑定到 n 个 100 Gbit/s 光模块进行传输。当接收端从光模块接收到信号后, 解除 FlexO 映射, 再将拆分的 n 个 OTU4 信号还原为一个 OTUC_n 信号。基于 FlexO 技术实现 400 Gbit/s 业务传输的原理如图 1 所示, 图中, CFP4 为 100 Gbit/s 光模块。

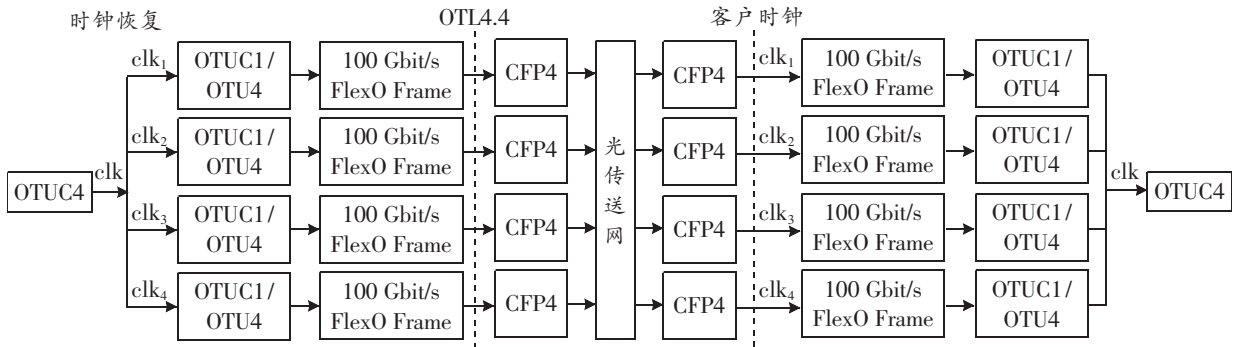


图 1 基于 FlexO 技术实现 400 Gbit/s 业务传输

400 Gbit/s 的 OTUC_n 业务数据拆分为多路 100 Gbit/s 的业务数据传输的过程, 是在时钟 clk 下完成的, 同时生成开销判断信号, 该信号可以指示时钟 clk 驱动下传输的每一拍数据, 是 OTU4 业务数据或是开销数据。此时需要建立 CDR 复模块, 滤除开销数据, 恢复每一路的客户时钟 clk_1 和 clk_2 等, 然后将重定时的 OTU4 业务数据通过相应接口送入光模块, 就可以实现 400 Gbit/s 的业务数据持续稳定向下一级传输。

本文即是在这样的应用场景下进行相应的 CDR 模块设计。其总体思路是利用异步先入先出 (First Input First Output, FIFO) 存储器实现 OTUC4 和 OTU4 两侧的数据速率区分。将 clk 作为 FIFO 的写时钟, 利用开销判断信号, 只把有效的 OTU4 数据写入 FIFO 存储器中。在读取端, 建立 CDR 算法, 该算法可以通过 FIFO 水位的提示, 自动调控锁相环频率合成器生成 FIFO 的读时钟。锁相环频率合成器的实现原理为

$$f_o = f_{\text{osc}}(1 + \Delta f \cdot 10^{-9}), \tag{1}$$

式中: f_{osc} 为晶振输入频率合成器的已知时钟; Δf 为待输出的读时钟 f_o 相对于 f_{osc} 的频偏量。建立的 CDR 算法本质上是一个采样反馈控制算法, 其采样周期为 T_s , 在 $t=0$ 时刻算法启动, 每隔 T_s 时间采样一次 FIFO 的水位值, 将该水位值与固定水位值的偏差作为算法的输入值。假设第 k 个采样时刻采集到算法的输入偏差为 $e(kT_s) = e(k)$, 按照其控制律算法可以输出锁相环频率合成器的频偏量 $\Delta f(k)$,

促使锁相环频率合成器按照式 (1) 生成读时钟 $f_o(k)$ 。CDR 算法的控制流程如图 2 所示。

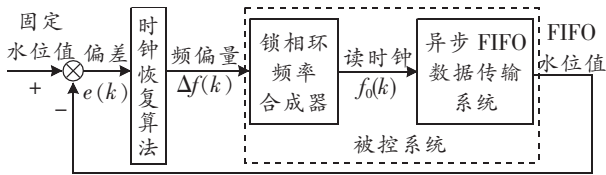


图 2 CDR 算法的控制流程

在经过多个采样周期后, 通过 CDR 算法的调整, 可以使 $e(k)$ 趋近于 0。此时, FIFO 的水位也保持为固定水位值不再发生变化, FIFO 两端的数据吞吐率相等, 在两端传输位宽相等的情况下, 由锁相环频率合成器生成的读时钟即为待恢复的客户时钟。由图 2 可知, 本文的 CDR 电路模块主要由 CDR 算法模块、锁相环频率合成器和异步 FIFO 模块 3 部分构成, 它与以往的时钟恢复电路执行的功能相同, 但结构形式上有较大差异, 其优点是可以引入各类优秀的控制类算法, 在优化 CDR 性能的同时, 可通过改善 CDR 算法实现, 而不用对电路结构形式作较大改动。

2 异步 FIFO 数据传输系统模型

根据前述的总体思路, 在进行 CDR 算法设计之前, 需要首先建立异步 FIFO 的数据传输系统模型。假设, 异步 FIFO 两侧的传输位宽相等均为 W , $t=0$ 时刻系统启动, 那么对于写入端, 其时钟为 f_i , 客户侧的数据吞吐率为 B_i , 那么对应的客户侧时钟为

$$f_c = B_i / W。 \quad (2)$$

第 n 拍写入 FIFO 的数据为 $F_i(n)$, 当第 n 拍数据为有效数据时, $F_i(n) = W$; 第 n 拍数据为开销数据时, $F_i(n) = 0$ 。规定 $F_i(1) = W$, 第 n 拍数据 $F_i(n)$ 相对于等效位宽 $w = B_i / f_i$ 出现的偏差 $e(n) = F_i(n) - w$, 对前 $n - 1$ 拍数据的偏差进行求和可得 $E(n - 1) = \sum_{j=1}^{n-1} e(j)$, 利用 $E(n - 1)$ 可以判断第 n 拍数据 $F_i(n)$ 的值, 当 $E(n - 1) < w$ 时, $F_i(n) = W$; 当 $E(n - 1) \geq w$ 时, $F_i(n) = 0$, 于是对应输入序列的模

表 1 不同数据吞吐率下开销数据的位置分布

数据吞吐 率/Gbit/s	开销序号										
	1	2	3	4	5	6	7	8	9	10	...
111.796 55	1 225	2 449	3 672	4 896	6 119	7 343	8 566	9 790	11 013	12 237	
111.803 26	1 322	2 642	3 963	5 283	6 603	7 924	9 244	10 565	11 885	13 205	
111.809 97	1 435	2 869	4 303	5 737	7 171	8 605	10 039	11 473	12 907	14 341	
111.816 68	1 570	3 139	4 708	6 277	7 845	9 414	10 983	12 552	14 121	15 689	
111.823 39	1 733	3 465	5 197	6 928	8 660	10 392	12 123	13 855	15 587	17 318	

对于读取端, 在第 n 拍时间内, 读取端的时钟为 $f_o(n)$, 可以得到输出序列的模型为

$$F_o(n) = W \cdot \frac{f_o(n)}{f_i}。 \quad (4)$$

若第 n 拍 FIFO 的水位为 $F(n)$, 则可得到异步 FIFO 的数据传输系统模型为

$$F(n) = F(n - 1) + F_i(n) - F_o(n)。 \quad (5)$$

CDR 算法的设计即是通过调整 $f_o(n)$ 使 FIFO 的水位稳定, 此时 $f_o(n)$ 将趋近于客户侧时钟 f_c 。读时钟 $f_o(n)$ 相对于客户侧时钟 f_c 的频偏为 $p(n)$, 满足 $|p(n)| > 1$ 的 n 的最大值记为 n_1 , 对应的 $\tau_1 = n_1 \cdot T_i$ 为 CDR 算法将读时钟调整为客户时钟的 $\pm 1 \times 10^{-6}$ 以内所需的时长, τ_1 越小则算法收敛速度越快, 对应 CDR 算法的速度就越快。算法执行较长时间后, 读时钟稳定后波动的最大值和最小值分别为 $f_{\max}$ 和 $f_{\min}$, 其对应的客户时钟的频偏分别为 $p_{\max}$ 和 $p_{\min}$, 令 $p = \max(|p_{\min}|, |p_{\max}|)$, 最终恢复的时钟在客户侧时钟 f_c 的 $\pm p$ 频偏范围内抖动, p 的值越小, 恢复的时钟抖动就越小。整个 CDR 过程中, 读时钟的最大值 $\max(f_o)$ 与晶振时钟 f_{osc} 的差值 $\Delta f_{\max}$ 为所需要的锁相环相对于 f_{osc} 上调的带宽, $\Delta f_{\max}$ 的值越小, 则所需要的锁相环的带宽越小。

CDR 的时钟恢复速度和恢复的时钟抖动是 CDR 电路重要的性能指标, 其值越小就表明 CDR 电路的性能越好。对于高速光通信系统而言, 其值

型为

$$F_i(n) = W \cdot \text{INT} \left[\frac{1}{2} \text{sgn} \left(\sum_{j=1}^{n-1} (F_i(j) - w) - W \right) + 1 \right], \quad (3)$$

式中: INT 函数为取整函数; sgn 函数为符号函数。本文选取的写时钟为 349.65 MHz, 位宽为 320 bit, 在不同数据吞吐率下, 利用 Matlab 软件确立开销数据的位置, 如表 1 所示。由表可知, 该模型方法可在客户侧特定数据吞吐率情况下, 使开销数据分布的尽可能均匀。

表 1 不同数据吞吐率下开销数据的位置分布

越小, 通信系统的稳定性就越高, 造成误码的概率就越低。锁相环带宽是从锁相环的角度来衡量 CDR 电路性能, 锁相环带宽较小的时候, 可以节省 CDR 电路中锁相环相关元器件的开销。

3 比例积分控制算法应用与验证

本文首先选取适合单输入单输出系统的数字比例积分 (Proportional-Integral, PI) 控制算法加以应用, 它是一个采样反馈控制算法, 以被控系统输出值 $y(k)$ 与参考值 $r(k)$ 的偏差 $e(k)$ 的比例和积分的线性组合, 输出调控量 $u(k)$, 其控制律为

$$u(k) = K_p e(k) + K_i \sum_{j=0}^k e(j), \quad (6)$$

式中: K_p 为比例增益系数; K_i 为积分增益系数。 K_i 对于调控量的改变影响较小, 而 K_p 对应的比例项反应系统的当前误差, 其值的选取对于调控量的改变影响较大。因此, 在应用 PI 控制算法时, 设置采样周期 $T_s = mT_i$, 将被控系统参数设置为 OTU4 业务频点的相关参数, 选取 $K_i = 1$ 不变, 然后通过改变 K_p 值得到读时钟 $f_o(n)$ 随时间变化的仿真结果, 所需要的参数如表 2 所示, 由式 (2) 可知, 客户侧时钟应为 349.407 9 MHz。PI 控制算法的调控, 即是将读时钟由 349.406 0 MHz 上调为 349.407 9 MHz。通过写时钟周期 T_i , 计算可得总的仿真时长约为 383.86 ms, 采样周期约为 5.86 μs 。

表 2 PI 控制算法参数

系统参数	数值	算法参数	数值
客户侧数据吞吐率/Gbit/s	111.810 53	仿真时长	$0\sim 2^{27}$
写时钟/MHz	349.65	标准水位	128
晶振输入时钟/MHz	349.406	采样周期 m	2^{11}
传输位宽/bit	320	K_i	1

本文选取 K_p 在 $10\sim 10\,000$ 之间,对 PI 控制算法的应用进行大量的仿真测试。当 $K_p < 20$ 时, τ_1 的值与仿真时长相等,此时调控量不足,收敛速度极慢,使得整个仿真时长内读时钟的值都没有收敛在客户时钟的 $\pm 1\times 10^{-6}$ 以内,如图 3(a) 所示。当 $20 < K_p < 100$ 时, τ_1 的值小于仿真时长,表明整个仿真时长内,PI 控制算法可以将读时钟收敛在客户时钟的 $\pm 1\times 10^{-6}$ 以内,如图 3(b) 所示。但此时调控量仍然不足,收敛速度较慢,随着 K_p 值的增加,算法的收敛速度在明显加快,对应恢复的时钟抖动也变得越来越大。 $K_p=100$ 为 PI 控制算法性能最佳的参数值,大约需要 103.01 ms 将读时钟收敛在客户时钟的 $\pm 1\times 10^{-6}$ 以内,需要锁相环能够相对于晶振输入时钟 349.406 MHz 上调带宽约为 3.469 kHz,最终恢复的时钟在客户侧时钟的 $\pm 0.122\times 10^{-6}$ 以内。当 $K_p > 100$ 时, K_p 的值继续增大,虽然算法的收敛速度在加快,需要锁相环的带宽在减小,但恢复的时钟抖动范围也在明显增大,恢复读时钟在尾部出现了高频跳变的“毛刺”现象, K_p 的值越大,“毛刺”的跳变范围就越大,如图 3(c) 和 3(d) 所示。这是由于 K_p 值较大时,PI 控制算法可以快速地将 FIFO 水位收敛在标准水位 128 附近,同时使读时钟接近客户时钟。但由于读时钟和客户时钟仍存在一定偏差,使得 FIFO 水位偏离标准水位 128,如 FIFO 水位由 128 变为 129,此时 FIFO 水位相比于标准水位 128 仅有一个单位的差值,但由式(6)处理后输出读时钟的调控量非常大,这就使得稳定后的读时钟在周期性地发生一定幅度的跳变。此时 PI 算法恢复的时钟频稳特性被“毛刺”现象制约,导致 PI 控制算法的其他性能被严重浪费。当 $K_p > 1\,000$ 时,恢复的时钟抖动范围已经超出了 1×10^{-6} ,所以在整个仿真时长内,PI 算法无法将读时钟调控在客户时钟的 $\pm 1\times 10^{-6}$ 以内。

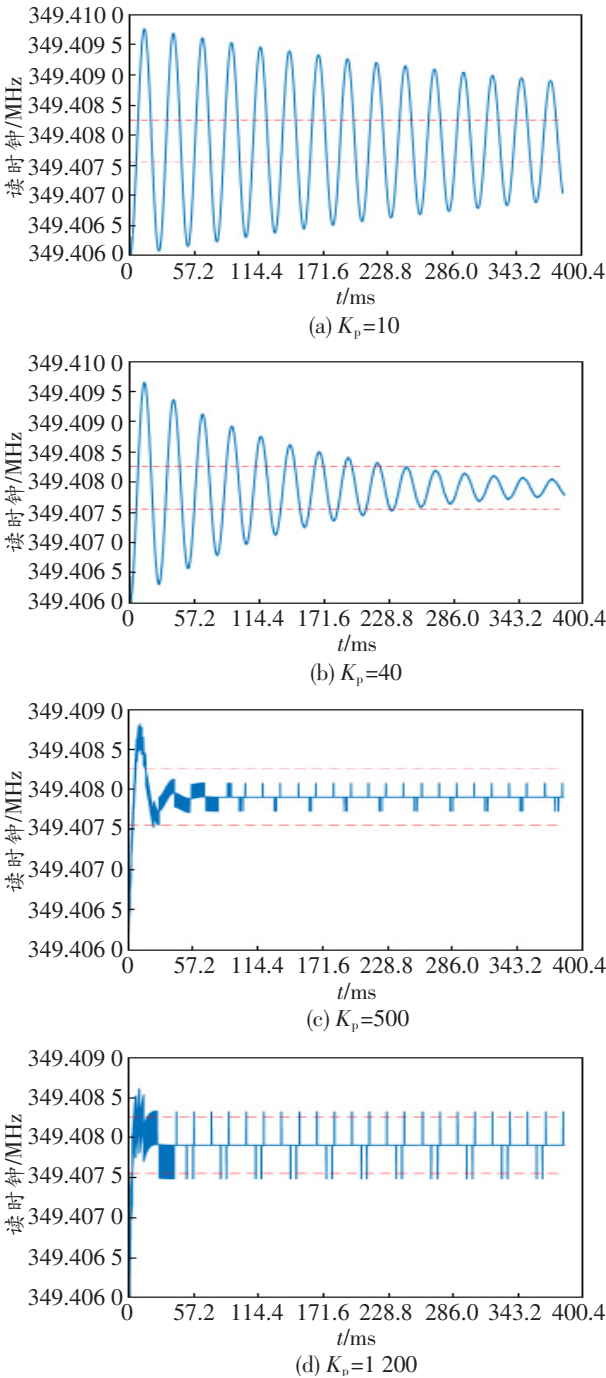


图 3 PI 算法调控生成读时钟的 Matlab 软件仿真结果(红色虚线为 $|p|=1$)

4 CDR 算法设计与验证

显然需要在 PI 控制算法的基础上,对式(6)给出的调控量产生一定的约束。假设从 $t=0$ 开始,第 k 个时刻(下文中时刻均为采样时刻),由式(6)输出 PI 控制算法的调控量为 $v(k)$,而实际输出的调控量为 $u(k)$,其偏差可以反映 k 时刻 FIFO 水位的状态:当 $|v(k)-u(k-1)|\geqslant u_s$ 时, k 时刻 FIFO 水位处

于不稳定状态;当 $|v(k) - u(k-1)| < u_s$ 时, k 时刻 FIFO 水位处于稳定状态;分别设置 p 和 q , 对 FIFO 水位的不稳定和稳定状态时刻进行计数, 当某个时刻 FIFO 水位处于不稳定状态时, 就使 p 的值加 1, 并令 q 的值为 0; 当某个时刻 FIFO 水位处于稳定状态时, 就使 q 的值加 1, 并令 p 的值为 0。

引入参数量 $S \in N^+$ (N^+ 为正整数集) 为 FIFO 水位特性判别时长, 若 k 时刻, $p=S$ 即 FIFO 水位连续 S 个时刻处于不稳定状态, 就认为 FIFO 水位处于震荡状态, 则开始震荡调控, 其调控模式如下:

$$\begin{cases} \text{当 } v(k) - u(k-1) > u_{\max} \text{ 时,} \\ \quad u(k) = u(k-1) + u_{\max}; \\ \text{当 } v(k) - u(k-1) < -u_{\max} \text{ 时,} \\ \quad u(k) = u(k-1) - u_{\max}; \\ \text{当 } |v(k) - u(k-1)| \leq u_s \text{ 时, } u(k) = v(k). \end{cases}$$

k 时刻调控结束后, 就使 p 的值减 1, 继续对 $k+1$ 时刻的 FIFO 水位状态进行采样计数。

若 k 时刻 $q=S$, 即 FIFO 水位连续 S 个时刻处于稳定状态, 就认为 FIFO 水位处于收敛状态, 则开始收敛调控, 其调控模式为

$$\begin{cases} \text{当 } v(k) - u(k-1) > u_{\min} \text{ 时,} \\ \quad u(k) = u(k-1) + u_{\min}; \\ \text{当 } v(k) - u(k-1) < -u_{\min} \text{ 时,} \\ \quad u(k) = u(k-1) - u_{\min}; \\ \text{当 } |v(k) - u(k-1)| < u_{\min} \text{ 时, } u(k) = v(k). \end{cases}$$

k 时刻调控结束后, 就使 q 的值减 1, 继续对 $k+1$ 时刻的 FIFO 水位状态进行采样计数。

u_s 为对 FIFO 水位状态进行鉴别时 $|v(k) - u(k-1)|$ 的门限值; $u_{\max}$ 为震荡调控下 $|\Delta u(k)|$ 的门限值; $u_{\min}$ 为收敛调控下 $|\Delta u(k)|$ 的门限值, 在选取参数值时应保证 $u_s \geq u_{\max} > u_{\min}$ 。

若 k 时刻 $p \neq S$ 且 $q \neq S$, 就认为 FIFO 水位处于过渡状态, 该状态持续的时间非常短暂, 此时继承 $k-1$ 时刻的调控模式。在调控系统刚刚启动时, 设置调控量 $u(0) = 0$, 将调控模式设置为震荡调控模式。然后基于 PI 控制算法输出调控量, 将整个 CDR 算法划分为震荡和收敛两个不同的调控模式。震荡调控模式下利用 $u_{\max}$ 对调控量进行约束, 收敛调控模式下利用 $u_{\min}$ 对调控量进行约束, 从而形成本文的 CDR 算法。

设置 $S=500$, $u_s = u_{\max} = 100$, $u_{\min} = 10$ 。同样地, 选取 K_p 在 $10 \sim 10\,000$ 之间, 对本文 CDR 算法的设计也进行仿真测试, 将不同 K_p 值对应的 τ_1 、

$\Delta f_{\max}$ 以及 p 与 PI 控制算法对比绘制成如图 4 所示的性能曲线。

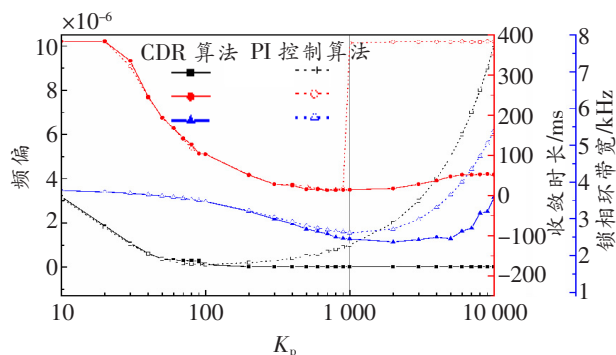


图 4 PI 控制算法和 CDR 算法的 K_p 性能对比

图 5 所示为 CDR 算法调控生成读时钟的仿真结果。当 $K_p < 100$ 时, 由式(6)输出的调控量较小, CDR 算法设置的门限对调控量几乎没有约束, 此时 CDR 算法和 PI 控制算法的性能基本一致, 如图 5(a) 和 5(b) 所示; 当 $K_p > 100$ 时, 在 CDR 算法调控下生成读时钟的仿真结果如图 5(c) 和 5(d) 所示, 此时由式(6)输出的调控量较大。但调控量受到了调控门限的约束, 在 CDR 算法调控的前期, FIFO 水位的偏差很大, 对应输出的调控量也很大, 此时 CDR 算法处于震荡调控模式, 限定了 $u(k)$ 的摆动幅度, 读时钟最大值减小, 对应所需要的锁相环带宽也减小。但由于震荡调控模式下的调控门限 $u_{\max}$ 较大, CDR 算法仍然可以输出较大的调控量, 促使读时钟快速收敛。之后 CDR 算法自动切换为收敛调控, 由于收敛调控模式下的调控门限 $u_{\min}$ 很小, CDR 算法只能输出很小的调控量, 从而确保生成的读时钟频具有较高的频稳特性。

由上述分析可知, 本文设计的 CDR 算法在选取 $K_p > 100$ 时, 不仅能够快速有效地恢复客户时钟, 而且恢复的时钟具有较高的频稳质量。CDR 算法的震荡调控模式给出较大的调控幅度, 相对于 PI 控制算法而言, 快速将时钟收敛的同时, 也降低了需要锁相环的带宽。之后进入收敛调控模式, 输出较小的调控幅度, 有效地抑制了 PI 控制算法在 $K_p > 100$ 时出现的“毛刺”现象。在 $500 < K_p < 2\,000$ 时, CDR 算法都拥有较高的性能, 此时约在 $13.47 \sim 21.61$ ms 就可以将读时钟调整为客户时钟 $\pm 1 \times 10^{-6}$ 以内, 需要锁相环的带宽在 $2.46 \sim 2.87$ kHz 之间, 恢复的时钟抖动约在客户侧时钟的 $\pm 0.02 \times 10^{-6}$ 左右。将该结果与 PI 控制算法的最佳性能对比可知, CDR 算法的性能有了较大提升。当 K_p 值继续增大时, 仍然能够在客户时钟的 $\pm 0.02 \times 10^{-6}$

左右恢复客户时钟,但收敛速度和锁相环带宽的性能在缓慢地变差,这是由于 K_p 值过大时,震荡调控下读时钟部分的瞬时值较大造成的。

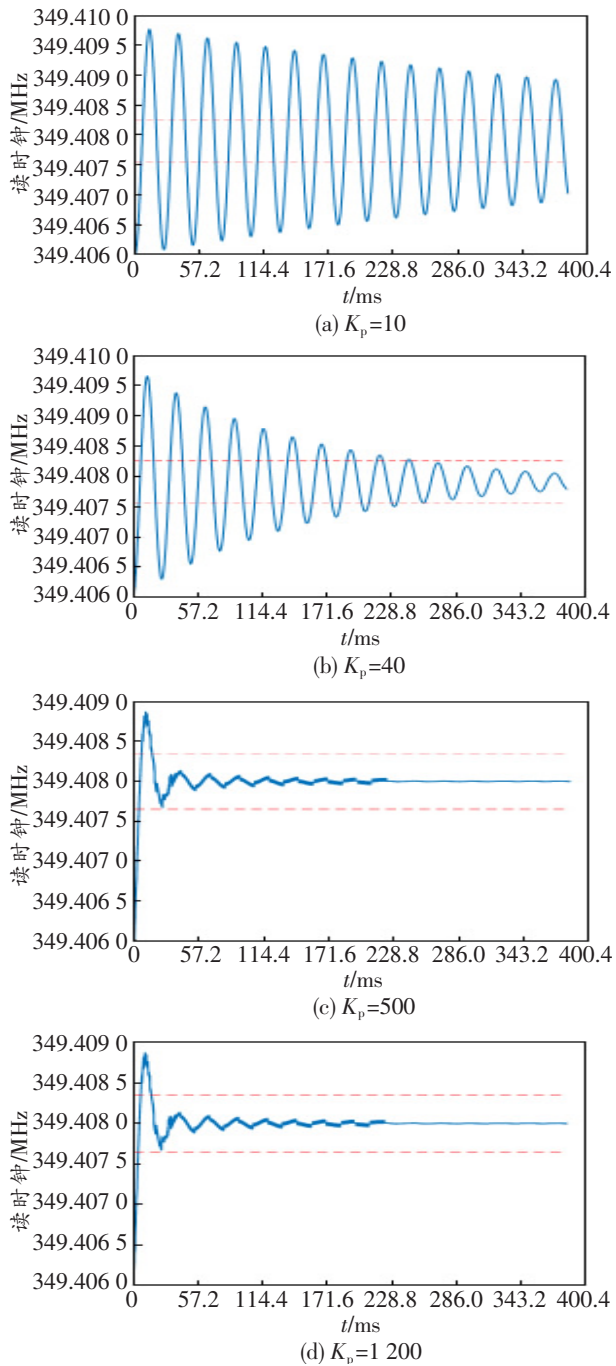


图 5 CDR 算法调控生成读时钟的 Matlab 软件仿真结果(红色虚线为 $|p|=1$)

在仿真获取算法的最佳性能参数后,本文选取不同的频偏值,在 $K_p=500$ 时,对 CDR 算法能够应对的频偏范围性能进行仿真测试,绘制的频偏性能曲线如图 6 所示。

本文设计的 CDR 算法对于 CDR 电路而言,在锁相环晶振输入的 349.406 MHz 的中心频率能够

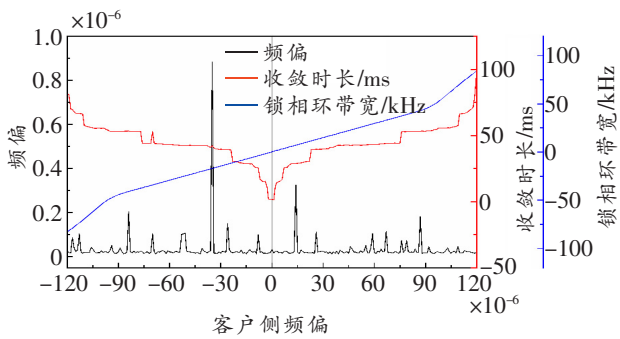


图 6 CDR 算法的客户侧频偏性能曲线

在 349.364 2~349.448 1 MHz 之间快速恢复时钟。对于高速光通信系统而言,在写入端相对读取端发生 $-120 \sim 120 \times 10^{-6}$ 的频偏,即数据吞吐率在 111.796 55~111.823 39 Gbit/s 之间,能够跟随写入端数据吞吐率的变化,实时恢复得到其对应的客户侧时钟,恢复的时钟相对于客户侧时钟的抖动在 $\pm 0.02 \times 10^{-6}$ 左右,完全能够满足基于 FlexO 的高速光传输系统的时钟恢复需求。一旦算法的数据吞吐率环境发生较大变化,仍可以通过改变算法参数来实现其他频偏范围的匹配适用。

5 FPGA 实现与实测验证

本文通过现场可编程门阵列(Field-Programmable Gate Array,FPGA)实现了 CDR 算法的模块设计,其实现框图如图 7 所示。将本文设计的 CDR 模块作为子模块例化在 OTUC4 向 4 路 OTU4 转化的模块中,并按照表 2 对所需参数进行配置,设置 $S=500,u_s=u_{\max}=100,u_{\min}=10$,搭建基于 FlexO 技术实现 400 Gbit/s 业务传输的测试环境。利用 100 Gbit/s 的 OTN 测试仪表将其中 1 路设置为 OTU4 的业务频点,相对于读取端的数据吞

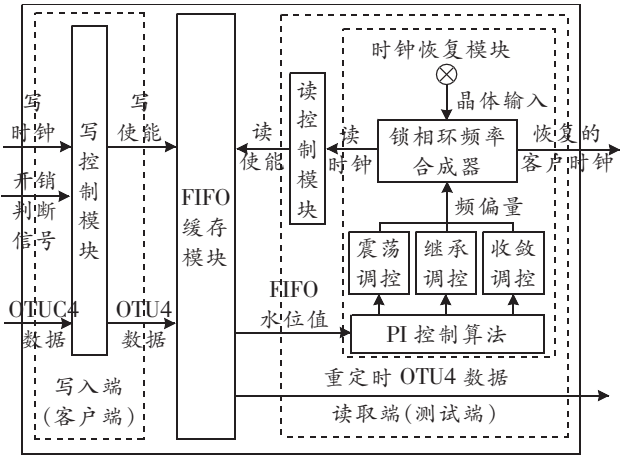


图 7 本文 CDR 模块的实现框图

吐率在写入端设置一定的频偏,然后观察读取端的数据吞吐率,以及是否出现误码。

其实测结果和算法结果的对比如表 3 所示。通过实测结果可以看到,基于本文 CDR 算法设计的时钟数据恢复模块可以有效满足 FlexO 技术中的 CDR 需求。恢复的客户侧时钟频偏不到 1×10^{-6} 。

表 3 FlexO 中 CDR 模块的实测结果与算法结果对比

频偏 $\times 10^{-6}$	客户侧数据 吞吐率/Gbit/s	客户侧 时钟/MHz	实测结果				算法结果			
			测试端数据 吞吐率/Gbit/s	恢复的 时钟/MHz	时钟频偏 $\times 10^{-6}$	是否 误码	频偏范围 $\pm 1\times 10^{-6}$	收敛 时长/ms	锁相环 带宽/kHz	
−90	111.799 937 1	349.374 80	111.799 911	349.374 72	0.233 5	否	0.020 6	53.04	−44.31	
−20	111.807 763 8	349.399 26	111.807 737	349.399 17	0.239 7	否	0.021 5	29.76	−9.66	
20	111.812 236 2	349.413 23	111.812 210	349.413 15	0.234 3	否	0.018 5	30.27	10.09	
91	111.820 174 7	349.438 05	111.820 148	349.438 04	0.238 9	否	0.020 4	53.59	45.67	
92	111.820 286 5	349.438 40	111.818 546	349.432 95	15.565 3	是	0.016 4	53.61	46.51	

6 结束语

本文设计实现了一种新型的全数字 CDR 电路,主要由异步 FIFO、锁相环频率合成器和 CDR 算法模块 3 部分构成,其核心是时钟恢复算法的设计。本文通过改进 PI 控制算法,设计实现了一种快速、低抖动的时钟恢复算法,并对算法的功能进行了仿真验证基于 FPGA 实现。仿真和实测结果表明,本文设计的 CDR 模块可以在 OTU4 业务频点的一定频偏情况内,实时、快速地在 1×10^{-6} 以内恢复客户时钟。通过分用 4 路或者复用 1 路本文设计的 CDR 模块,就可以满足 400 Gbit/s 超大带宽业务传输的 CDR 需求。本文所提方案对于 CDR 电路结构改进也具有一定的借鉴意义。

参考文献:

[1] 曾庆珠,高桂香. 400 Gbit/s 光传输技术研究[J]. 微波学报, 2016, 32(4):92−96.

[2] 郭俊. 高速 SerDes 系统的时钟恢复电路设计研究[D]. 杭州:浙江大学, 2017.

[3] Jeong C, Abullah A, Min Y, et al. All-Digital Duty-Cycle Corrector with a Wide Duty Correction Range for DRAM Applications[J]. IEEE Transactions on Very

实测结果中,客户侧频偏在 92×10^{-6} 时,CDR 模块失效,出现误码行为,这是由于锁相环相对于 349.406 MHz 上调的带宽不足 46.51 kHz 造成的。实际上,若锁相环带宽性能足够,CDR 模块能够在 111.796 55~111.823 39 Gbit/s 之间实时恢复客户时钟。

Large Scale Integration (VLSI) Systems [J]. 2016, 24(1):363−367.

[4] 李志贞. 超高速并行时钟数据恢复电路的研究与设计[D]. 南京:南京邮电大学, 2016.

[5] 吕建新,谢秋红,刘红捷,等. 100 Gbit/s 以太网关键技术研究及实现[J]. 光通信研究, 2012(1):1−5.

[6] 吕建新. 超高速光通信的新技术及应用[J]. 电信科学, 2011, 27(11):127−133.

[7] 胡腾飞. 一种基于锁相环与延迟锁相环混合结构的时钟数据恢复电路设计[D]. 合肥:中国科学技术大学, 2016.

[8] 唐玉红. PID 控制方法研究[J]. 电子世界, 2019(7): 65−66.

[9] ITU-T G. 709/Y. 1331-2016. Interface for the Optical Transport Network[S].

[10] 吴飞权. 高速 ADC 时钟占空比较准电路设计[D]. 合肥:合肥工业大学, 2016.

[11] 伍得阳. 低抖动时钟占空比较准电路的研究与设计[D]. 上海:复旦大学, 2013.

[12] Liang J, Sheikholeslami A, Tamura H, et al. A 28 Gbit/s Digital CDR with Adaptive Loop Gain for Optimum Jitter Tolerance [C]//IEEE International Solid-State Circuits Conference. San Francisco, CA, USA:IEEE, 2017:122−123.