A Robust Architecture Based on Adaptive Recursive Filter for Gigabit Communications

Kothapalli Roopa, R. V. Siva Krishna Addagattu, and Kuppili Sunil Kumar

Abstract This paper proposes a novel design which is capable of faster data recovery for high-speed communications such as serializer and deserializer (SerDes). The SerDes is an essential and widely used component in high-speed gigabit communications such as PCIe, Ethernet, passive optical networks (PON), MIPI and HD video streaming interfaces. The high-speed serializer/deserializer is the dominant implementation of I/O interfaces at speeds of 2.5 Gbps and higher. The SerDes works with a source-synchronous interface in which no synchronization clock will be present while transmitting/receiving the data. The SerDes receivers must have the clock and data recovery (CDR) circuit which dynamically extracts the clock and data from the receiving differential serial data. The design in this paper proposes a Nyquist sampling-based architecture that will simultaneously capture the serial data with high redundancy and without any bit loss. The architecture uses an algorithm which also features the adaptive sampling rate independent of the bit duration. The algorithm is capable of estimating the interleaving window between successive bits and significantly analyzes the samples of successive bits and dynamically filters the noisy samples and recovers the bit information and also has the ability to adjust the offset deviations occurred while sampling the serial data. The algorithm is implemented and verified on the SerDes serial receiver at 25 Gbps data rate at 14 nm technology node.

Keywords SerDes · LVDS · PON · CDR · Channel noise · SNR · BER

K. Roopa (✉)
Department of Electronics and Communications Engineering, Raghu Institute of Technology (RIT), Visakhapatnam, India
e-mail: roopa.kothapalli@gmail.com

R. V. Siva Krishna Addagattu
Qualcomm Technology India Pvt. Ltd., Bangalore, India

K. Sunil Kumar
Dr. B. R, Ambedkar University, Srikakulam, India

P. S. R. Chowdary et al. (eds.), *Microelectronics, Electromagnetics and Telecommunications*, Lecture Notes in Electrical Engineering 655,
https://doi.org/10.1007/978-981-15-3828-5_1

1 Introduction

1.1 High-Speed Digital Communication

The chip density is increasing day by day, and the demand for the high-speed circuits has skyrocketed. Previously, high-speed data communication was meant to be for limited applications and offered at higher prices. As the chip density increases day by day, the larger complex circuits can be accommodated in the integrated circuits which brought the high-speed circuits [1] into daily life for almost all each and every application. This has increased the demand for the circuits which can perform high-speed gigabit communications. Eventually, the design requirements and design challenges, pushing the limits of the current technology and reducing the time to market, have become the inevitable task for the engineers.

Low-Voltage Differential Signaling (LVDS)
At first glance, it might appear that one of the disadvantages of utilizing LVDS in an application as opposed to a customary single-finished information transmission technique is that it requires double the same number of wires to transmit a similar number of channels. In contrast, the inverted pair of wires used in the place of single wire will help us to increase the data speed [2]. The two wires for the same signal will carry bit information inverted to each other. This will help to recover the data at the receivers when being transmitted at gigabits per second. In reality, the LVDS interface can undoubtedly reduce the wires between the transmitter and receiver for higher data speeds. On the contrary, the throughput will be the same when compared with the multiple bits sent via the parallel interface at lower data rates and data sent serially using LVDS at higher data rates.

The LVDS can be implemented for multiple configurations as shown in Fig. 1. The point to point configuration for end-to-end interface and multibit configuration to interface multiple devices is significantly more efficient unlike bus interface. The multiple channels can be utilized for LVDS interface to increase the overall throughput. The LVDS is being utilized in major high-speed applications such as mobile industry processor interface (MIPI), high-definition media interface (HDMI), modem interface and PCIe.

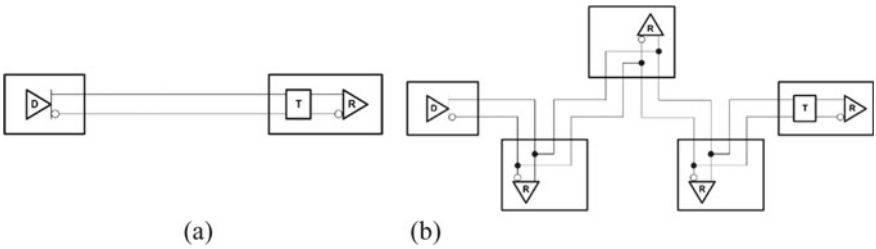

Fig. 1 LVDS **a** point-to-point configuration, **b** multidrop configuration

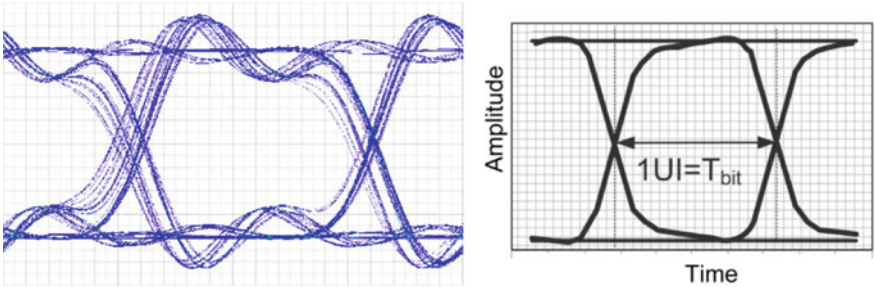

Fig. 2 Eye diagram of high-speed differential signaling

The LVDS is likely to be more redundant to environmental noise and demonstrated robustness toward common-mode noise and also less affected by the noise-related problems such as cross talk from the neighboring channels. This will also enable the use of LVDS in low-power applications with higher data rates. Eye diagrams are used to characterize a high-speed signal source or transmitter (receiver testing usually requires bit error rate testing). A typical eye diagram test setup is shown in Fig. 2. Eye amplitude is the difference between the one and zero levels. The data receiver logic circuits will determine whether a received data bit is a "0" or "1," based on the eye amplitude.

The vertical opening is the main characteristic while recovering the bit information from the differential signals which define the eye height of an eye diagram. The channel noise, cross talk, signal-to-noise (SNR) ratio, jitter and additional noise parameters will influence the eye amplitude. All these noise parameters will be involved and try to reduce the eye amplitude which will cause the eye to close in high-speed communications. If the eye amplitude reduced, it is very difficult to recover the clock and data information using CDR. This paper addresses this problem and proposes a possible solution to recover the data under these eye closing circumstances.

2 Theory

2.1 SerDes *Overview*

The basic architecture of the *SerDes* is shown in Fig. 3. The architecture is mainly consisting of parallel in serial out (PISO) and serial in parallel out (SIPO) shift register to convert the parallel data into serial and serial data into parallel, respectively [3]. The architecture of *SerDes* can be classified into four categories (1) parallel clock *SerDes*, (2) bit interleaved *SerDes*, (3) embedded clock *SerDes* and (4) 8b/10b SerDes which is exclusively used for passive optical network (PON) applications.

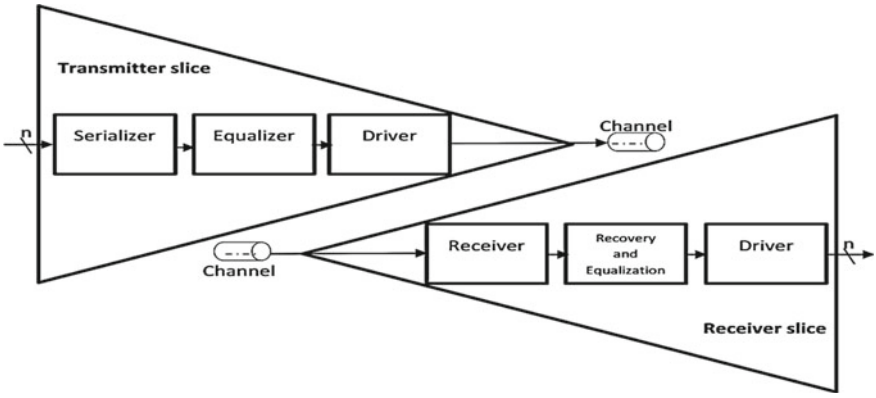

Fig. 3 Architecture of *SerDes*

The parallel in serial out (PISO) is typically consisting of parallel interface with the number of flip flops to hold the data, clock input for the flip flops and a serial output line where the serial data is available. The PISO module utilizes the internal or external phase-locked loop (PLL). The basic element in parallel in serial out (PISO) is a shift register which will convert the parallel data into serial out for one bit per clock.

However, the design will require various synchronizing schemes when the data is transferred from one clock domain to another clock domain to avoid metastability and also to prevent the data loss.

The serial in parallel out (SIPO) performs an opposite function to parallel in serial out (PISO). The serial in parallel out (SIPO) used in *SerDes* slightly differs from conventional SIPO. The SIPO used in *SerDes* will not be given any the same clock which was given at the PISO as there will be no reference clock provided along with the data. The *SerDes* internal clock management system will recover the clock information from the serial data after analyzing the serial bits, and then, the CMS will provide the corresponding reference clock to the phase lock loop (PLL). The PLL will lock the reference clock and generate the serial clock which will be given to serial in parallel out (SIPO) shift register. Further, the SIPO will shift the serial high-speed data and then converts it into parallel bit information. The clock management system (CMS) and phase-locked loop (PLL) performance should be good enough so that the low harmonic frequencies in the data stream and frequency offset can be nullified.

2.2 Clock and Data Recovery (CDR)

In the cutting-edge high-speed data circuits, specifically the gigabit transceivers, solid impedances and cutoff between intersymbol interference (ISI), is completely

dependent on frequency. The intersymbol interference (ISI) is considered as one of the major causes for performance degradation of theses gigabit transceivers by effecting in eye opening and timing jitter which leads to data mismatch [4] and bit error rate (BER). The major challenge in the gigabit communications is that there is no dedicated clock is allotted to the signal in high-speed SerDes transceivers. The clock and data recovery (CDR) plays a vital role [5] in gigabit communications. As there will not be any reference clock sent along with the high-speed serial data, the CDR must analyze the serial bit pattern and extract the clock information from the incoming serial data. Clock and data recovery (CDR) units are observing the transitions and choosing the minimal sampling phase for the data at the midpoint between edges. It retrieves clock data from the actual receiving data stream and uses this extracted clock to recollect the data waveform and absorb the data. CDR is nonlinear which significantly limits jitter and noise inside the SerDes [6] circuit. There is numerous clock recovery circuit (CRC) design approaches existing like traditional CRC, over-sampled CRC, source-synchronous links, etc. where the data stream must guarantee transitions. Most of all clock and data recovery circuits employ phase-locked [7] loop circuits.

3 Proposed Design

This paper proposed a Nyquist sampling-based architecture that will simultaneously capture the serial data with high redundancy and without any bit loss. The architecture uses an algorithm which also features the adaptive sampling rate independent of the bit duration. The algorithm is capable of estimating the interleaving window between successive bits and significantly analyzes the samples of successive bits and dynamically filters the noisy samples and recovers the bit information and also has the ability to adjust the offset deviations occurred while sampling the serial data. The block diagram of the design is shown in Fig. 4. The serial data is sampled by the sampler and collects a fixed number of samples for each serial bit. The sampler in the design is independent of the bit duration and also scalable. This scalability feature does not limit the number of samples to be fixed.

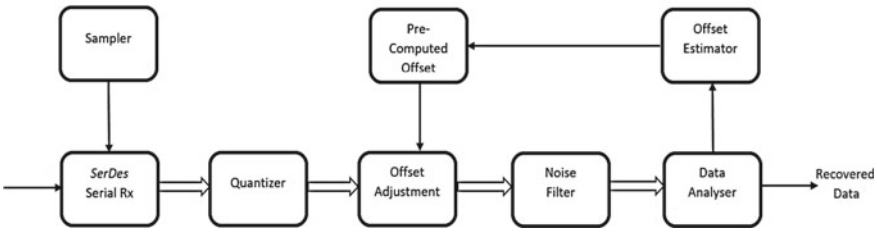

Fig. 4 Block diagram of proposed design

The number of samples required to sample the data can be varied from one scenario to another scenario considering several factors such as bandwidth, signal-to-noise ratio (SNR), bit error rate (BER) and noise present in the data. The sampling frequency should be significantly higher than the speed of the incoming data. The sampling frequency of the serial data can be chosen depending upon the number of samples that need to be collected for bit analysis in the noisy medium. The major feature of this architecture is that it will be suitable for high-speed source-synchronous interfaces where no reference clock will be distributed along with the serial data. The quantizer collects the sampled information of each bit and then identifies the samples related to transitional window and active window where the actual portion of the samples will be utilized for data recovery.

The transitional window samples are used to adjust offset deviations between the successive bits while sampled by the sampler. This technique effectively tunes the number of transitional samples required for each bit depending on the offset deviations caused by the sample. This will help the design to be more robust. The quantizer categorizes the samples into two groups such as (i) *transitional window* and (ii) *active window*.

Transitional Window

The transitional window is the key area that will compute the offset caused by the residual components while sampling the information. This transitional window further subdivided into pre- and post-computed offset windows. The pre-computed offset window will hold the previous bit offset deviation which will affect the sampling duration of the current bit. This bit-to-bit offset will be summarized and results in data mismatch due to the offset residual components [8]. The post-computed offset window which was computed successfully after the previous data will be buffered as a pre-computed offset to utilize during the next bit sampling window for tuning the transitional window. This will dynamically tune the complete sampling window for successive bits when they were retrieved at higher data rates.

The pre- and post-computed window size is scalable and can be chosen at which the rate of data being transmitted and offset deviation between the residual components generated while sampling (Fig. 5).

The post-computed offset will be taken into account when the next serial data is getting sampled, and the pre-computed window and number of samples collected in the active window will be adjusted dynamically such that offset deviation will be

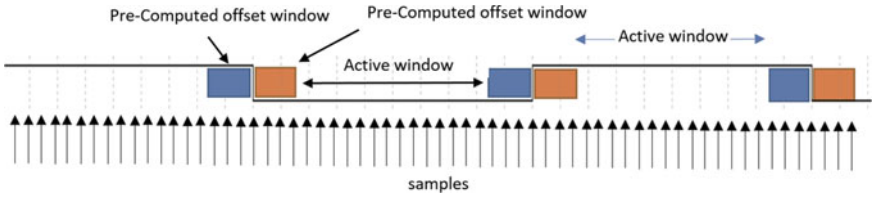

Fig. 5 Sampling methodology

nullified. In this way, the proposed design will address the offset deviation between successive bits during the high-speed reception.

$$m = M - \left(\Delta m_{\mathrm{p}} + \Delta m_{\mathrm{q}}\right) \tag{1}$$

$$\begin{aligned} \text{rxd\_int}\,(n+1) = {} & '1' \text{when} \sum_{k=0}^{m-1} x(n-k) \geq \tfrac{M}{2} \\ & '0' \text{ otherwise} \end{aligned} \tag{2}$$

$$\begin{aligned} \text{rxd\_int}\,(n) = {} & '1' \text{ when} \sum_{k=0}^{m-1} x(n-k) \geq \tfrac{M}{2} \\ & '0' \text{ otherwise} \end{aligned} \tag{3}$$

where

m	Active window samples
M	Total number of samples collected for differential serial bit
Δm_{p}	Pre-computed offset values
Δm_{q}	Post-computed offset values.

Active window
The active window is the group of samples that will be used to extract the bit information. The high-speed interfaces will communicate in differential signaling (LVDS). Thus, both positive and negative samples will be combined together. Then, the samples will be processed further to recover the bit information. The noise filter will be filtering the non-differential samples which should be a compliment in nature. Then, the inverted samples will be processed, and the final bit information will be evaluated after analyzing both inverted samples collected from the active window. This will enhance the robustness of the design which enables the design for better performance and faster data recovery.

4 Results

The proposed design is successfully implemented on a *SerDes* receiver at the data speed 25 Gbps. The SerDes has 40-bit parallel input bus operating at 625 MHz which will match the throughput of the serial data 25 Gbps. Similarly, it has the 40-bit parallel output data bus where the received serial data is latched. The reference clock of 156.25 MHz will be supplied to the PLL for generating the internal transmitter and receiver clocks. The *SerDes* serial receiver while receiving the data at 25 Gpbs data speed, each bit has duration of 37.253 ps. Each bit was sampled at 1 ps sampling clock resulting in 37 samples per bit (Fig. 6).

The 0.253 ps residual cumulatively will create an offset of 25 ps after receiving every 100 bits. This 25 ps offset is significantly capable of data corrupt or mismatch.

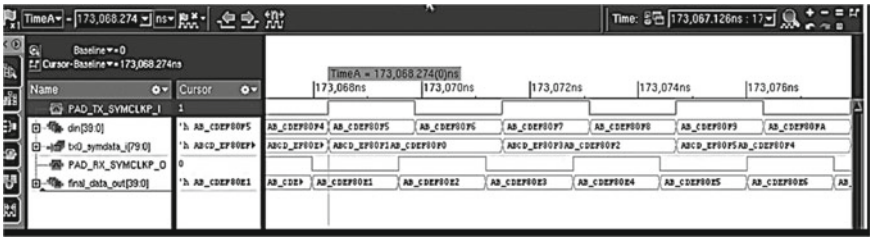

Fig. 6 SerDes recovered data with the proposed algorithm

To avoid this, the above algorithm was implemented on the receiver samples. The algorithm was dynamically self-tuned and adjusted the offset of each consecutive bit, and the offset of 0.253 ps per bit was successfully nullified without inserting delay.

5 Conclusion

This paper focuses on challenges that arise while receiving high-speed serial data only as it has more complex issues unlike transmission of the data. The SerDes works with a source-synchronous interface in which no synchronization clock will be present while transmitting/receiving the data. The SerDes receivers must have the clock and data recovery (CDR) circuit which dynamically extracts the clock and data from the receiving differential serial data. The algorithm is capable of estimating the interleaving window between successive bits and significantly analyzes the samples of successive bits and dynamically filters the noisy samples and recovers the bit information and also has the ability to adjust the offset deviations occurred while sampling the serial data. This will dynamically tune the complete sampling window for successive bits when they were retrieved at higher data rates.

References

1. Chang E, Narevsky N, Han J, Alon E (2018) An automated SerDes frontend generator verified with a 16NM instance achieving 15 GB/S at 1.96 PJ/Bit. In: Symposium on VLSI circuits, 2018. Honolulu, IEEE, pp 153–154
2. Gupta HS, Parmar R, Dave R (2009) High speed LVDS driver for SERDES. In: International conference on emerging trends in electronic and photonic devices and systems. IEEE, Varanasi, pp 92–95
3. Hou C, Wang Z, Huang K, Zhang C, Wang Z (2014) A 20 GHz PLL for 40 Gbps SerDes application with 4-bit switch-capacitor adaptive controller. In: International conference on electron devices and solid-state circuits. IEEE, Chengdu, pp 1–2
4. Stauffer DR, Mechler JT, Sorna MA, Dramstad K, Ogilvie CR, Mohammad A, Rockrohr JD (2008) High speed SerDes devices and applications. Springer, USA

5. Razavi B (2002) Challenges in the design of high-speed clock ad data recovery circuits, communications magazine. IEEE, pp 94–101
6. Zhihui Z, Yuan W, Junlei Z, Hailing Y, Song J (2010) A clock and data recovery circuit for 3.125 Gb/s RapidIO SerDes. In: International conference of electron devices and solid-state circuits (EDSSC). Hong Kong, IEEE, pp 1–4
7. Prinzie J et al (2019) A low noise fault tolerant radiation hardened 2.56 Gbps clock-data recovery circuit with high speed feed forward correction in 65 nm CMOS. In: 10th Latin American symposium on circuits & systems (LASCAS). m IEEE, Armenia, Colombia, pp 63–66
8. Roosevelt G, Roper W, Romanko T (2011) Optimizing high speed serial communication using Honeywell Rad Hard SerDes. In: NASA/ESA conference on adaptive hardware and systems (AHS). San Diego, CA, pp 215–219