MPLEG: A Multi-mode Physical Layer Error Generator for Link Layer Fault Tolerance Test

Xingyun Qi(✉), Pingjing Lu, Jijun Cao, Yi Dai, Mingche Lai, and Junsheng Chang

College of Computer, National University of Defense Technology, Changsha, Hunan, China
qi_xingyun@mail.nudt.edu.cn

Abstract. In the design of high-speed communication network chips, the fault-tolerant design of the link layer is among the most important parts. In the design process, the link layer fault tolerance function need to be fully tested and verified. But it is far from enough to rely only on traditional case-by-case simulation. In order to test and verify this function completely, this paper proposes a configurable multi-mode physical layer error generation method implemented on chip: MPLEG (a Multi-mode Physical Layer Error Generator). With MPLEG, a desired bit error pattern can be generated at the physical layer in all stages of chip design, including simulation verification, FPGA prototype system verification, sample chip testing, and actual system running. The statistical analysis of the experimental results shows that MPLEG can generate an error pattern almost identical to the real link error. Meanwhile, MPLEG can perform relatively complete and efficient testing and verification of various functions of link layer fault tolerance.

Keywords: Physical link error · Error pattern analyze · Error generator on chip · Link layer fault tolerance test

1 Introduction

In high-speed digital communications, the traditional parallel transmission method has been unable to meet the ever-increasing communication bandwidth requirements, and high-speed serial transmission methods have emerged. In the serial communication mode, digital communication between two nodes is performed through high-speed serial differential signals. The transmitter encodes the parallel data and clock signal, and then converts it into a serial data stream through a SERializer/DESerializer (SERDES), and then sends it to the physical medium; the SERDES in the receiving end receives the serial data, then the serial data is converted into parallel data, and the clock of the receiving end is recovered by the clock and data recovery (CDR) circuit. According to the network layer structure, serial-to-parallel conversion, clock recovery, data encoding and decoding, and data transmission processes in serial digital communications all belong to the physical layer function [1]. In this process, the high-speed serial signal may cause errors in the transmission process due to various reasons. Under normal circumstances, there may be two reasons for the occurrence of bit errors: (1) due to the timing jitter, the actual signal transition is earlier or later than the ideal position; (2) due to amplitude noise,

D. Dong et al. (Eds.): ACA 2020, CCIS 1256, pp. 56–69, 2020.
https://doi.org/10.1007/978-981-15-8135-9_5

the high level of the signal is lower than the reference threshold or the low level of the signal is higher than the reference threshold. Therefore, in a high-speed communication system, after the link layer receives data from the physical layer, one of the important functions is the fault tolerance processing, including data verification, error detection, error correction, and retransmission. The purpose of fault tolerance is to provide reliable data transmission services for the upper layers of the network.

In the design of communication systems, especially the design of high-speed network communication chips, in order to test the correctness of the link layer error processing system, it takes a lot of resources to verify and evaluate each function of the fault tolerance system. In the process, the most critical thing is to artificially generate various errors close to the real situation at the physical layer, so as to cover most error testing scenarios. The physical layer errors generated for the purpose of performing link layer fault-tolerant tests play a vital role in the correctness and completeness of the final system verification.

There are currently two commonly used link fault tolerance test methods: simulation test and real system test. The simulation test method is using VCS [2]/Xcelium [3] and other front-end simulation tools to simulate the RTL design, and manually inserting the errors expected by the designer on the link. But the test data of this method is limited, and the subjective bias of the designer may not be able to reproduce the error scenario under the real environment, and its test coverage is limited. The real system test method is to build a real hardware test environment, apply interference on the physical link, and then test the fault tolerance design of the system. The real system test method can indeed reproduce the real error scenario, but the cost is too high, and it is not easy to find the error. At the same time, the real system test method generally needs to be carried out after the chip has been taped out, so it is difficult to perform a complete test at the design stage. Therefore, to design a link fault tolerance test method which is capable of simulating real physical link errors inside the chip is necessary, so as to carry out various boundary and abnormal tests on link transmission in the design stage of high-speed communication chips.

Aiming at the above problems, this paper first collects the bit errors on the real link, analyzes its distribution characteristics, and designs an on-chip error generation method of physical layer - MPLEG according to the characteristics. Using MPLEG, a physical layer error similar to the real situation can be generated inside the high-speed communication chip, which is used to test the fault-tolerant design inside the chip.

The rest of this paper is organized as follows: Sect. 2 introduces the relevant research background; Sect. 3 collects and analyzes the bit error characteristics on the actual physical link; Sect. 4 presents the method of MPLEG; Sect. 5 evaluates the effect and performance of on-chip error generated by MPLEG; Sect. 6 summarizes the work and draws relevant conclusions.

2 Related Works

Most researches about physical link error pattern are focused on wireless communications [10–14] and free space optical links [15]. In [10], the target packet error sequences are generated by a simulator with a typical urban (TU) channel and co-channel interference. A general design procedure of a generative model is then proposed by using a properly parameterized and sampled deterministic process with a threshold detector and two parallel mappers. A generative error model that can generate packet-level error sequences with predicted burst error statistics is proposed in [11], which can generate errors similar to those of error sequences obtained from real wireless systems. [12, 13] both proposed a link error prediction method respectively. Both methods are shown to have accuracy within a few tenths of a dB under a wide range of modulation schemes, coding rates and channel types. These methods are then extended to handle more advanced link enhancements such as hybrid ARQ and Alamouti encoding [14]. Wu et al. [16] studied chip error patterns in IEEE 802.15.4, and found out that there are four major error patterns. They proposed a simple yet effective method based on the chip error patterns to infer the link condition.

For high speed serial communication, there are currently two methods for evaluating the signal quality: simulation analysis and worst-case link analysis. Among them, the simulation analysis method generally uses tools such as SPICE for pseudo-random data simulation, which usually uses a Pseudo-Random Bit Sequence (PRBS) generator to simulate the link channel. The worst-case link analysis method calculates the worst eye diagram of the system by analyzing the channel response and noise model, and estimates the worst bit error rate of the system [4–6]. Both of these two methods have advantages and disadvantages. The simulation analysis method needs to generate a large amount of random test cases to cover various transmission scenarios of the link, therefore it is difficult to ensure complete coverage of various test scenarios [8]. The worst-case analysis method only focuses on the link in the worst case. Therefore, the result is too pessimistic, which will put more strict requirements on the design [7]. Some researchers [8] studied the error conditions of high-speed links from the perspective of probability statistics, and proposed a new bit error rate analysis method to analyze the eye diagram of any pattern under the influence of link ISI and crosstalk. Then the paper calculated the probability of each situation, and obtained the bit error rate eye diagram of the link according to the probability distribution at the receiving end. Dongwoo Hong et al. believes that the error in high-speed communication mainly comes from data jitter, including random jitter (RJ) and deterministic jitter (DJ) [9]. This paper proposes a method to theoretically analyze the RJ and DJ of signals in the data at the receiving end, and gets the bit error rate on high-speed serial links.

Some researchers assumes that the ideal noise sources follow the Gaussian distribution. For real noise source generation, two uniformly distributed pseudo-random signals are generated, which are then transformed into actual noise sources through function transformation, and implemented on the FPGA chip. However, this method is only compared with the theoretical Gaussian distributed noise source, but not with the actual noise on the real link.

All the researches above have only analyzed and estimated the noise on the physical link. Some have only designed an ideal noise source, but they have not been compared with the noise on the actual link. Moreover, most of the current researches focus on the theoretical analysis of the noise and interference on the serial link. These researches lack statistics and analysis of interface data error between physical layer and link layer, which is precisely the link layer fault tolerance design should pay attention to. Some researches

This paper is geared towards the fault-tolerant design of the link layer. It collects and analyzes the true bit errors of the data on the interface between the physical layer and the link layer. Based on the analysis, a physical layer bit error generator inside the chip is proposed, which can generate various modes of bit error close to the actual bit error mode. It can be used to fully test the fault-tolerant design of the link layer.

3 Statistics of Real Bit Error in Physical Links

In order to analyze the real bit error characteristics on the physical link in the high-speed serial communication, we use an FPGA chip evaluation board with a high-speed serial communication interface to design a hardware system that can collect and analyze the physical layer data transmission bit errors. The system communicates between two FPGA chips through a high-speed serial interface. The transmitter continuously sends out pseudo-random sequences as test data, while the receiver continuously receives data and checks the correctness of the data. The system structure is shown in Fig. 1.

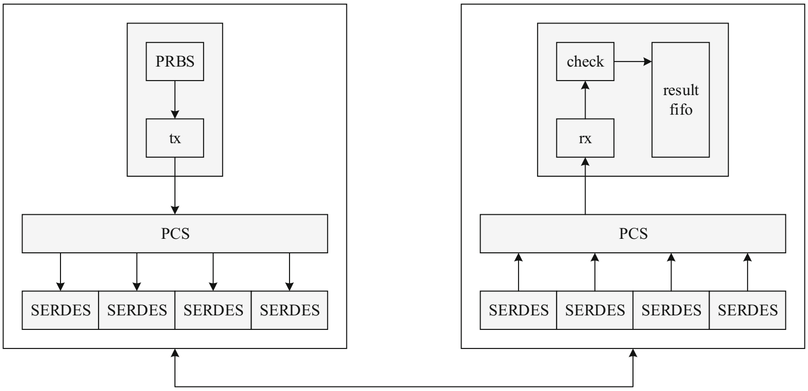


Fig. 1. The structure of the physical layer error capture

At the receiving end, each received data is checked for correctness. If data error occurs during transmission, the receiver will record the specific error information, including the expected data, the wrong data, the time interval between this error and the previous error, and the total number of error bits. Based on these information, the distribution characteristics of error can be figured out, and the error model can be obtained.

In the experiment, we chose a fiber with errors with which the two FPGA are connected in the test system. The FPGA chips choosed in the experiment are Xilinx Virtex 7 2000T devices. In the evaluation board, each port uses a 4lane GTX, with the rate of 1.25 Gbps per lane. The test system based on FPGA evaluation boards is illustrated in Fig. 2.

The test continues about 30 min, and the data is continuously sent at the maximum rate. The receiver detects the correctness of the data and records the error information. The test results are shown in Table 1.

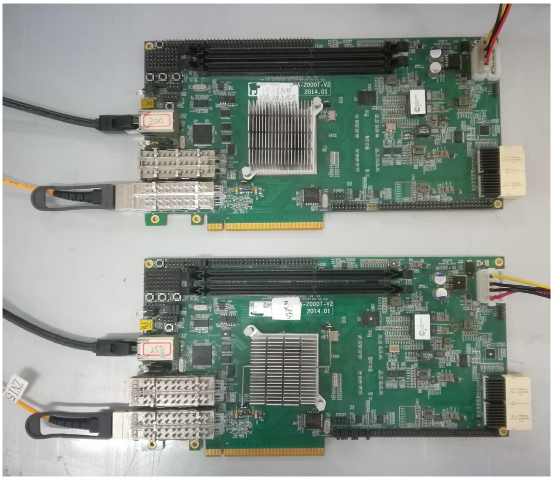


Fig. 2. Link error test system based on FPGA

Table 1. Statistics of physical link real error information

Items	Value
Total amount of the test data	7.5×10^{12} bits
Total error bits	134128 bits
Average error rate	1.788×10^{-8}
Average interval of 2 error (take 256bit data as the unit)	4.4706×10^{5}

We analyzed the time interval between two adjacent errors, and take statistics of the interval distribution. Figure 3(a) is a distribution diagram of different error intervals. It can be seen that the number of errors with an interval of 1 (that is, two adjacent errors) is significantly larger than the errors of other intervals. The number of errors of other intervals gradually decreases with the increasing of interval. We can see that adjacent error with an interval of 1 is the burst error. Figure 3(b) is the error distribution diagram with the error interval greater than or equal to 2, and Fig. 3(c) is the error distribution diagram with the error interval of 1. It can be seen from the figure that the bit errors in the actual link are actually the superposition of burst errors and uniform errors. That is, Fig. 3(a) is the superposition of Fig. 3(b) and Fig. 3(c).

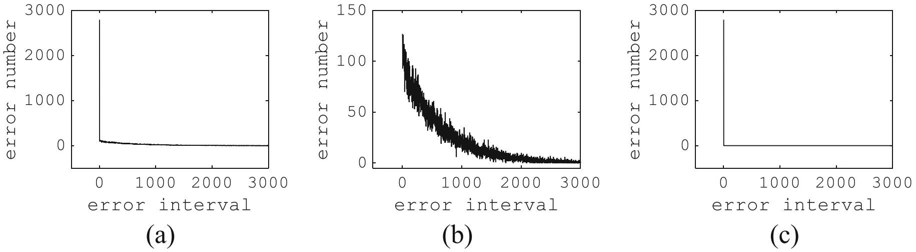


Fig. 3. Number of errors in different intervals

4 Multi-mode Physical Layer Error Generator - MPLEG

By analyzing the bit error characteristics of the real physical link in above experiment, we find that the bit error on the actual link is a mix form of uniform bit error and burst bit error. According to this feature, we design a configurable multi-mode physical layer error generator - MPLEG. The error generator is located between the PCS (Physical Coding Sublayer) and SERDES at receiver, and is used to inject a specific pattern of error sequences into the data from SERDES. According to this structure, an error generating module (error_insert) is designed at the receiving end of each lane to insert a specific error bit sequence into the parallel data given by SERDES, as shown in Fig. 4.

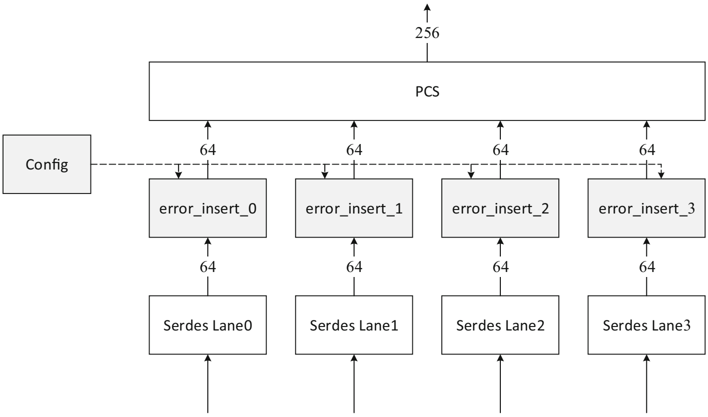


Fig. 4. Structure of error generator

In Fig. 4, there are 4 lanes in each network port. After serial-to-parallel conversion, each SERDES provides 64-bit-wide parallel data bits to PCS. The 4 lanes are bound together to provide a 256-bit-wide data path. Under the control of the Config module, each error_insert module reverses every bit in the incoming 64-bit data according to certain error generation rules. For example, if the link error rate is set to 10^{-8}, then the probability of error per bit is $1/10^{8}$. That is, in that lane, a bit flips once about every 108 clock cycles on average (0 to1 or 1 to 0). Each error_insert contains 64 error-generating units (named err_element), and each error-generating unit generates errors in a certain bit of 64-bit data. Therefore, all data bit errors are independent of each other, as shown in Fig. 5.

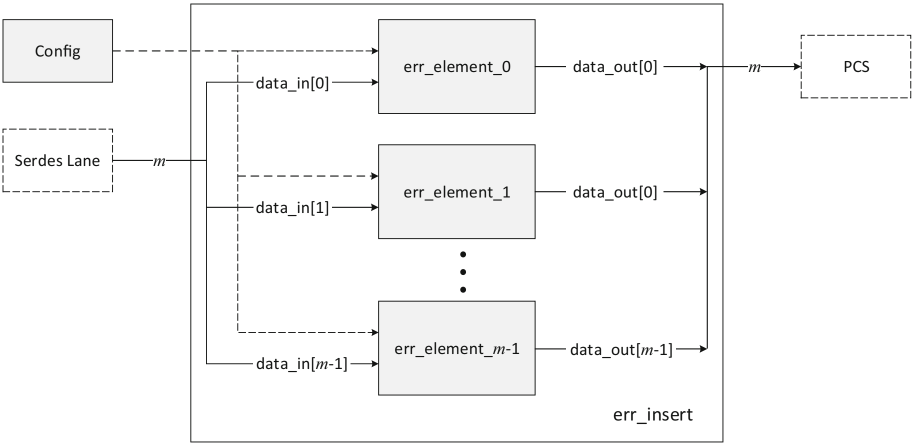


Fig. 5. Structure of error_insert module

The function of the error-generating unit is to generate all kinds of error patterns according to the configuration. Based on the previous analysis of the error characteristics on the physical link, the error-generating unit should provide multiple configurations to achieve a variety of error distribution characteristics such as uniform error, burst error and mixed error. From the perspective of functional components, the error-generating unit is divided into a uniform error generator, a burst error generator, and a synthesizer, which are used to generate uniform errors, burst errors, and multiple error modes. The uniform error generator generates uniform errors according to the input signal cfg_err_rate. The error rate of the uniform error is cfg_err_rate/232. The burst error generator generates two kinds of error with different error rates (high error rate and low error rate) according to the input signals cfg_err_burst_low_period, cfg_err_burst_high_period, cfg_err_rate_low and cfg_err_rate_high. The high error rate is cfg_err_rate_high/232 and the low error rate is cfg_err_rate_low/232, and the durations of high error rate and low error rate are cfg_err_burst_high_period and cfg_err_burst_low_period, respectively. In this way, when different configuration are given, different burst errors can be generated. According to the configuration, the synthesizer combines the previously generated uniform errors and burst errors into a mixed error mode, and the physical layer error is the result of a mixture pattern of uniform errors and burst errors. The structure of error-generating unit is shown in Fig. 6.

The processing flow chart of error-generating unit is shown in Fig. 7.

When the configuration signal cfg_err_mode is 0, no error is generated, that is, output data data_out is equal to input data data_in. When cfg_err_mode is 1, errors are generated in uniform mode, that is, the output data data_out is the inverse of the input data data_in with certain probability. When cfg_err_mode is 2, the error-generating units

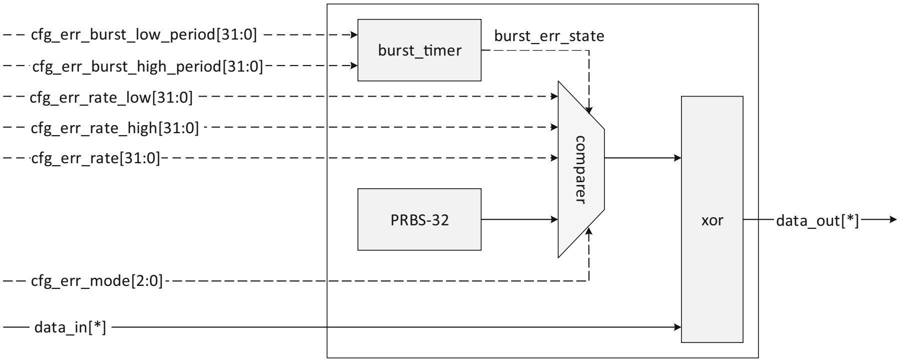


Fig. 6. Structure of error-generating unit

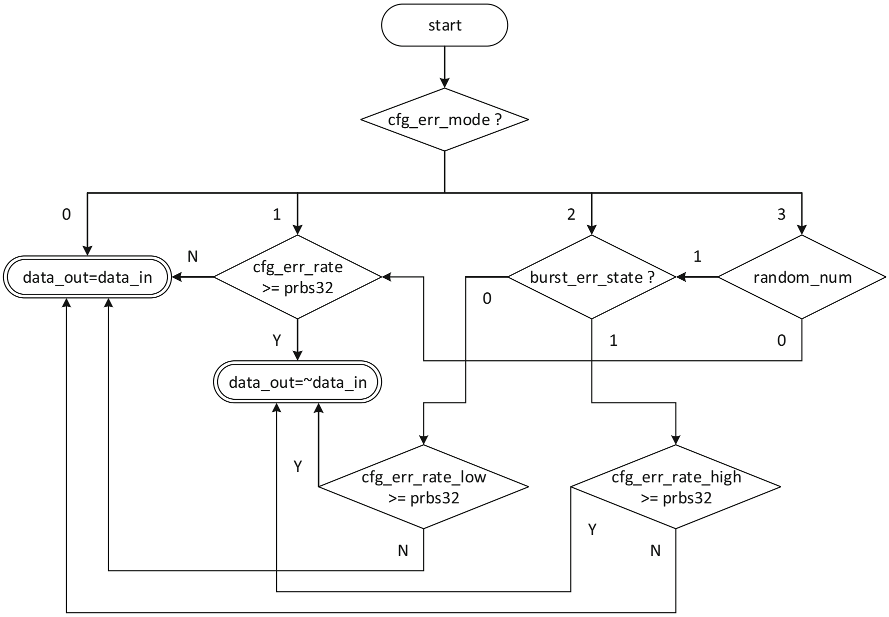


Fig. 7. The processing flow of error-generating unit

are in burst error mode, that means, two different periods are generated according to the configuration signal burst_time, which are corresponding to the high error rate time and low error rate time. In the high error rate period, the error rate is cfg_err_rate_high, and in the low error rate period, the error rate is cfg_err_rate_low. That is, there exists bit error in output data data_out compared with data_in. The error rate changes between two different preset bit error rates according to the configuration. In this way, the burst error mode can be achieved. When cfg_err_mode is 3, the error-generating units are in mix mode. In this mode, a 1-bit random number (random_num) is generated to determine whether it is currently in uniform error mode or burst error mode randomly. When the random number is 0, it is in a uniform error mode, and when the random number is 1, it is in a burst error mode.

In this way, an error generating module (error_insert) contains 64 independent error-generating units, and each error-generating unit generates an error to one of the data bit. Figure 8 shows the structure of an error_insert module.

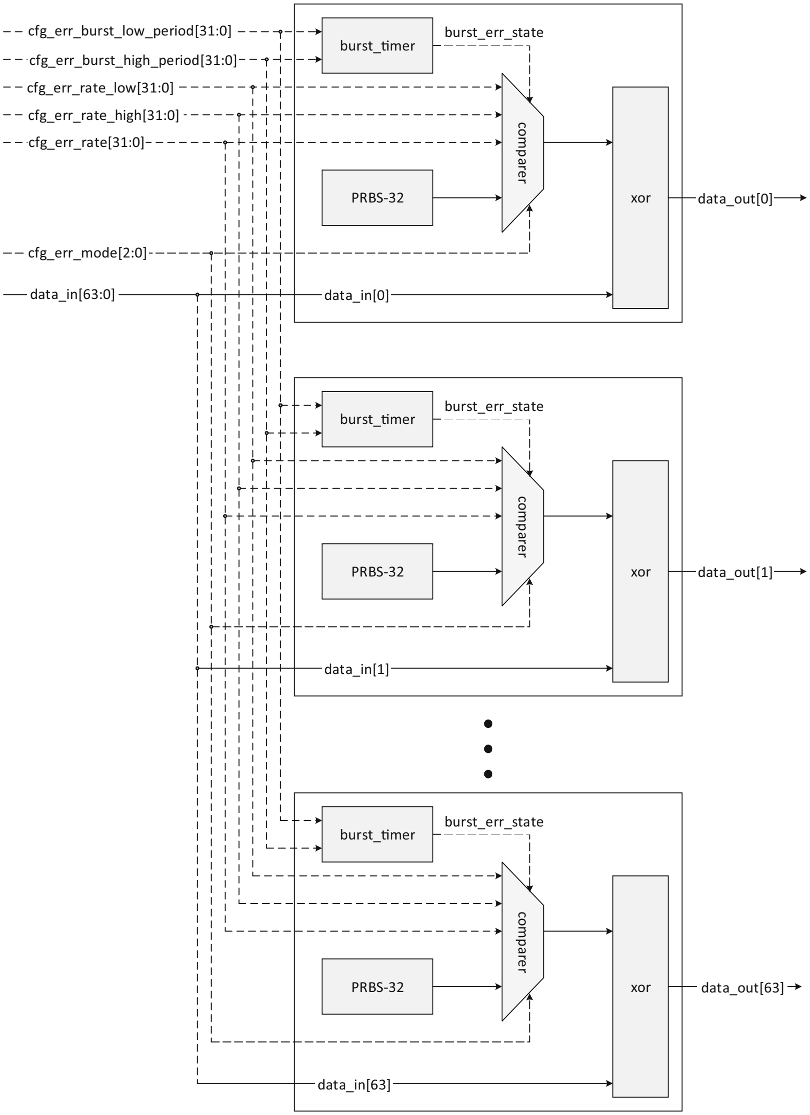


Fig. 8. Structure of an error_insert module

5 Evaluation

We will analyze and evaluate the errors at the physical layer generated by the physical layer error generator proposed above.

5.1 Evaluation Criterion

The error data recorded in the experiment is an n-tuple, where n is the number of records. In the experiment, we take $n = 65535$. Therefore, the actual error data is recorded as

$$R = [r_1, r_2, \cdots, r_{65535}]$$

The error data generated by the error generator described in this paper is recorded as

$$G = \left[g_1, g_2, \cdots, g_{65535}\right]$$

Where $r_i, g_i = 0, 1, 2, \cdots$, means the interval between the current error data and the last error data.

For example, if a certain error data is recorded as [74, 382, 0, 1673, 258, 4, 67], it means that a total of seven errors occurred during the recording time. There are 74 correct data between the first error data and the first data, and 382 correct data between the second error data and the first error data, and 0 correct data between the third error data and the second error data (which means the second and the third error data are continuous), …, and 67 correct data between the seventh error data and the sixth error data.

In order to evaluate the difference between R and G, we define the distribution distance between two sets of data records below.

Let $s = 1, 2, 3, \ldots$, and s is the dividing scale. Divide the record R and G according to the following algorithm, and calculate their distribution distance:

(1) Let $m = \max\{R, G\}$

(2) Initialize 2 $\left\lfloor \frac{m}{s} \right\rfloor$-dimensional vectors

$$V_r\left[0 : \left\lfloor \frac{m}{s} \right\rfloor - 1\right] = 0$$

$$V_g\left[0 : \left\lfloor \frac{m}{s} \right\rfloor - 1\right] = 0$$

(3) for $i = 1, 2, \ldots, 65535$

$$V_r\left[\left\lfloor \frac{r_i}{s} \right\rfloor\right] = V_r\left[\left\lfloor \frac{r_i}{s} \right\rfloor\right] + 1$$

$$V_g\left[\left\lfloor \frac{g_i}{s} \right\rfloor\right] = V_g\left[\left\lfloor \frac{g_i}{s} \right\rfloor\right] + 1$$

(4) When the division scale is *s*, the average distribution distance between *R* and *G* is defined as

$$D_s(R, G) = \left| \frac{1}{\lfloor \frac{m}{s} \rfloor} \sum_{i=0}^{\lfloor \frac{m}{s} \rfloor - 1} \left(V_r[i] - V_g[i] \right) \right|$$

and the accumulated distribution distance between *R* and *G* is defined as

$$E_s(R, G) = \frac{1}{\lfloor \frac{m}{s} \rfloor} \sum_{i=0}^{\lfloor \frac{m}{s} \rfloor - 1} \left| V_r[i] - V_g[i] \right|$$

When $s = 1$, V_r and V_g indicate the number of error messages with different error intervals. That is, $V_r[0]$ represents the number of 0s in *R*, i.e. the number of error packets with an error interval of 0. $V_r[1]$ represents the number of 1s in *R*, i.e. the number of error packets with an error interval of 1.

When $s > 1$, it is equivalent to dividing the data in *R* and *G* into $\lfloor \frac{m}{s} \rfloor$ groups respectively with the interval *s* from small to large. Then the number of data in each group are counted. Thus, $V_r[i]$ indicates the number of data whose value is $i \cdot s \sim (i+1) \cdot s - 1$ in *R*. For example, $V_r[0]$ represents the number of data in *R* with a value of 0 ~ $s - 1$, that is, the number of error packets with an error interval of 0 ~ $s - 1$. $V_r[1]$ represents the number of data in *R* with a value of *s* ~ $2s - 1$, that is, the number of error packets with an error interval of *s* ~ $2s - 1$.

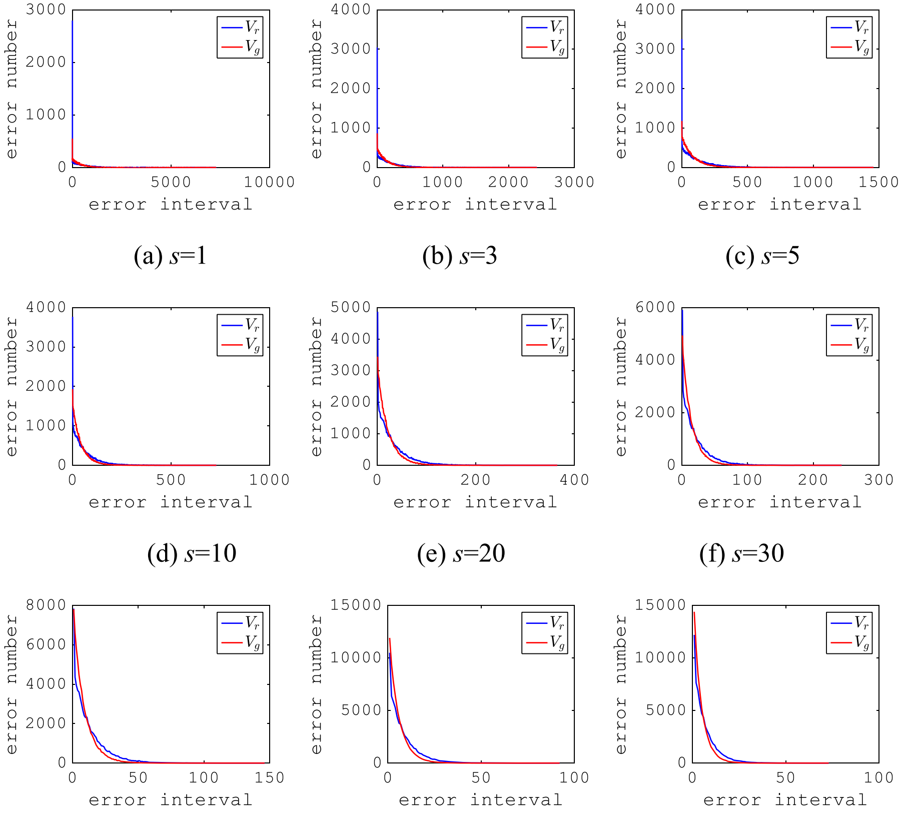
When *s* is constant, $D_s(R, G)$ and $E_s(R, G)$ is the distance between vectors *R* and *G*. The smaller $D_s(R, G)$ and $E_s(R, G)$ is, the smaller the difference between *R* and *G* is.

5.2 Evaluation of Generated Error Data

We use the FPGA test system in Sect. 3 to evaluate the error data generated by the physical layer error generator. Firstly, the FPGA system is tested with an optical fiber that has bit errors, and a set of original error record *R* with the elements number of 65535 is obtained. Then the record *R* is analyzed to calculate the bit error rate. Based on this error rate, appropriate parameters of the error generator are set. Then the FPGA system is tested again using a normal fiber. The error results of the test are recorded as *G*. According to the analysis method mentioned before, the difference between *R* and *G* is evaluated for the two sets of error records.

According to the algorithm presented in 5.1, we take different division scales $s = 1$, 2, 3, …, 100 to analyze *R* and *G* respectively. Figure 9 shows the curves of V_r and V_g when *s* is 1, 3, 5, 10, 20, 30, 50, 80 and 100. That is, the distribution of *R* and *G* under different division scales *s*. It can be seen from the figure that the distribution of the error interval of *R* and *G* are basically coincident. It indicates that the result of MPLEG are close to the result of actual physical link errors in different scale *s*, and the physical layer errors generated by MPLEG can basically fit the error pattern under actual conditions.

We also calculated the distance between actual error record *R* and generated error record *G* at different division scales ($s = 1$, 2, 3, …, 5000). Figure 10(a) and (b) show the curves of $D_s(R, G)$ and $E_s(R, G)$ with the increasing of *s*, respectively. It can be seen from the figure that no matter what the value of s, the curve of $D_s(R, G)$ is basically 0. That is, the average distance between R and G remains very low, which indicates that the distribution curves of R and G have significant similarities. At the time, $E_s(R, G)$ increases with the increase of s. When $s > 1000$, $E_s(R, G)$ gradually decreases with the increase of s. Under different values of s, the average value of $E_s(R, G)$ is 684.5, which is still at a low level, indicating that the difference between R and G is small.



(a) s=1

(b) s=3

(c) s=5

(d) s=10

(e) s=20

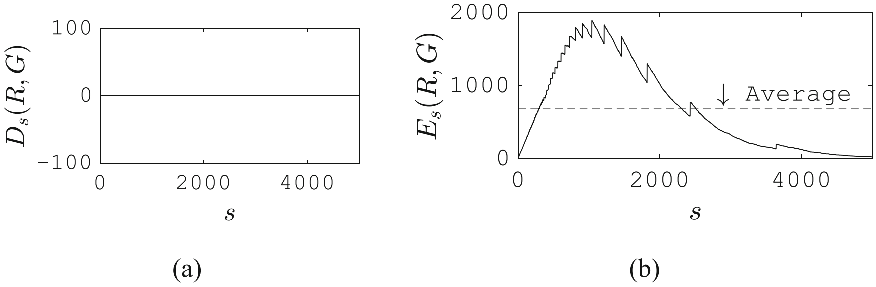
(f) s=30

(g) s=50

(h) s=80

(i) s=100

Fig. 9. V_r and V_g in different s



(a)

(b)

Fig. 10. Distance between *R* and *G*

5.3 Actual Link Layer Fault Tolerance Test

The purpose of MPLEG in this paper is to facilitate the test of the fault tolerance in the link layer. In order to verify whether the function of MPLEG proposed in this paper could test the fault tolerance function of the link layer comprehensively and reasonably, we use fiber with poor link quality and normal fiber with the error generator to test a same network environment respectively. Then we compare the performance differences of link layer fault tolerance in these two different environments. Table 2 shows the number of link layer retransmissions per second in the two error scenarios under different bit error rates.

Table 2. Link layer retransmission number in different error rate

Error mode	Number of error rate			
	$3 * 10^{-9}$	$2 * 10^{-8}$	$1 * 10^{-7}$	$3 * 10^{-6}$
Actual physical layer error	147	282	1218	14038
MPLEG	164	298	1176	14357

It can be seen from the table that MPLEG has an impact on the fault tolerance processing of the link layer, which is basically the same as the impact of the physical layer errors under the real link on the link layer. Under the two methods, the numbers of retransmissions in the link layer are basically equal within a certain period of time. So the transmission qualities of the two situation are same from the view of link layer. We can simulate the bit error scenario on the real physical link and test the link layer fault tolerance function using the physical layer error generation method in this paper.

6 Conclusion

This paper aims at the difficulty of effectively making errors on the physical link during the design and verification process of the current high-speed interconnection network chip, which makes it impossible to fully test and verify the fault tolerance function of the link layer. According to the actual error characteristics of the link, a configurable multi-mode physical layer error generation method - MPLEG is proposed. Using MPLEG, a configurable error mode that combines burst errors and uniform errors can be generated based on actual link error patterns and scenarios. By comparing the actual physical link error with the generated error, we find that MPLEG is similar to actual link layer error in many aspects such as error mode, error distribution characteristics and impact on the actual link layer fault tolerance processing. MPLEG can simulate the errors on the physical link in the chip, and provide strong support for the test of the link layer fault tolerance function.

References

1. Tanenbaum, A.S, Wetherall, D.J.: Computer Networks. 5 edn. (2010)
2. Synopsys VCS. https://www.synopsys.com/verification/simulation/vcs.html. Accessed 21 Apr 2020
3. Cadence Xcelium Logic Simulation. https://www.cadence.com/en_US/home/tools/system-design-and-verification/simulation-and-testbench-verification/incisive-enterprise-simulator.html. Accessed 21 Apr 2020
4. Casper, B.K., Haycok, M., Mooney, R.: An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes. In: IEEE Symposium on VLSI Circuits Digest of Technical Papers, Honolulu, HI, USA (2002)
5. Casper, B., O'Mahony, F.: Clocking analysis, implementation and measurement techniques for high-speed data links—a tutorial. IEEE Trans. Circ. Syst. **56**(1), 17–39 (2009)
6. Ren, J., Oh, D.: Multiple edge responses for fast and accurate system simulations. IEEE Trans. Adv. Packag. **31**(4), 741–748 (2008)
7. Chada, A.R., Wu, S., Fan, J., et al.: Efficient complex broadside coupled trace modeling and estimation of crosstalk impact using statistical BER analysis for high volume. high performance printed circuit board designs. In: IEEE 63rd Electronic Components and Technology Conference (ECTC), Las Vegas, NV, pp. 2095–2101 (2013)
8. Li, Y.: BER performance analysis of high-speed parallel link. Master's Thesis of XiDian University (2015)
9. Hong, D., Ong, C.-K., Cheng, K.-T.: Bit-error-rate estimation for high-speed serial links. IEEE Trans. Circ. Syst.—I: Regul. Pap. **53**(12), 2616–2627 (2006)
10. Wang, C-X., Xu, W.: Packet-level error models for digital wireless channels. In: Proceeding of IEEE ICC 2005, Seoul, Korea, pp. 2184–2189, May 2005
11. Salih, O.S., Wang, C-X., Mesleh, R., Ge, X., Yua, D.: Predicting burst error statistics of digital wireless systems with HARQ. In: 9th International Wireless Communications and Mobile Computing Conference (IWCMC) (2013)
12. Lampe, M., Rohling, H.: PER-prediction for PHY mode selection in OFDM communication systems. In: Proceeding of IEEE GLOBECOM 2003, vol. 1, pp. 25–29, December 2003
13. Ericsson: System-level evaluation of OFDM - further considerations. 3GPP TSG-RAN WG1#35, R1–031303, pp. 17–21, November 2003
14. Blankenship, Y.W., Sartori, P.J., Classon, B.K., Desai, V., Baum, K.L.: Link error prediction methods for multicarrier systems. In: Proceeding of VTC 2004-Fall, Los Angeles, USA, pp. 4175–4179, September 2004
15. Ansari, I.S., Yilmaz, F., Alouini, M.-S.: Performance analysis of free-space optical links over Málaga (M) turbulence channels with pointing errors. IEEE Trans. Wirel. Commun. **15**(1), 91–102 (2016)
16. Wu, K., Tan, H., Ngan, H., Liu, Y., Ni, L.M.: Chip error pattern analysis in IEEE 802.15.4. IEEE Trans. Mob. Comput. **11**(4), 543–552 (2012)