Journal of Physics:
Conference Series

PAPER • **OPEN ACCESS**

An 8B/10B parallel encoder design for the polarity pre-processing

To cite this article: Siqi Song *et al* 2023 *J. Phys.: Conf. Ser.* **2450** 012055

An 8B/10B parallel encoder design for the polarity pre-processing

Siqi Song[1,2*], Heguo Yin[1], Jianzhong Zhao[2], Yi Zhan[2] and Peter Poechmueller[1]

[1]School of Microelectronics, Shandong University, Jinan, Shandong, 250101, P.R.China

[2]Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, 10029, P.R.China

[*]Corresponding author's e-mail: 202012442@mail.sdu.edu.cn

Abstract. To tackle the high latency and slow encoding speed of 8B/10B encoder circuits in high-speed serial interfaces, a parallel structure for polarity pre-processing based on logic operations is proposed to optimize encoder performance in this article. Firstly, the encoding process is divided into polarity pre-processing and coding two pipeline stages. This method equals each stage's delay and decreases the coding waiting period. Additionally, the structure employs polarity pre-processing to optimize the critical delay paths into multi-stage heterodyne gates in series. Therefore, the effect of the number of input bytes on the critical delay path can be significantly reduced. The effectiveness of our proposed encoder is demonstrated to encode 4-byte loads by VCS simulation and synthesized using Synopsys' Design Compiler tool upon the SMIC 65nm process. Results show that a high operating frequency of 909MHz with a 2245.68μm^2 footprint area can be achieved. Compared with the cascaded and polarity-selective structures, the proposed framework has advantages in terms of speed and resource usage. These advantages greatly help to meet the high-speed transmission requirement.

1. Introduction

In serial data transmission systems, there is no strict limit to the number of "0" and "1". When consecutive, the voltage fluctuates, which can seriously affect the signal's integrity and cause electromagnetic interference to the system[1]. Albert X. Widner proposed the 8B/10B coding method and Peter A. Franaszek in 1983, which allows 8bit data in any data mode to be coded as 10bit data without D.C. components[2]. Due to a series of advantages such as high conversion density, easy clock recovery, easy, fast synchronous alignment, and D.C. balancing, 8B/10B encoding is widely used in high-speed serial communications such as PCIe, SATA, and Fiber Channels.

8B/10B encoding is generally implemented by the lookup table method or logic operation method, among which the lookup table method is the most commonly used 8B/10B encoding implementation method due to its simplicity and low encoding error rate[3]. However, the continuous development of modern electronics and communications technology requires data transmission systems with much higher transmission speeds. The lookup table method can no longer meet the needs of modern communications at high speed, which is constrained by the RAM reading speed, slow operation speed, and occupies large resources[4], and has been improved to solve those problems. By modifying the lookup table, designs have been developed to run up to 342MHz under the SMIC180 process[5] and 350MHz under Cyclone IV[6]. By combining logic operations with the lookup table, designs have been developed to run up to 614MHz under the ZC706 development board[7] and up to 1.6GHz quad-equivalent on Arriva V[8]. But the single-byte coding speed of such designs still fails to meet the 1GHz

application requirements. In the literature [1], a 2GHz encoder design was implemented in a 0.18μm process by using a combination of logic operations and lookup tables, with a special structure for the flip-flop and MOS circuit [1]. The design has a high area of 5080 μm^2 and consumes 9.38 mW of power, making it overly resource-intensive and under-versatile.

Compared to lookup table methods, the logical operation method is becoming the dominant coding method in high-speed serial communications with the advantages of being faster and less resource intensive[3]. In the current research on the implementation of single-byte 8B/10B encoders for logic operations, encoders have been designed for 1GHz operation at 65nm and 90nm[4,9]. In contrast, the complexity of the coding logic leads to severe jitter in logical operation, which limits the further increase in coding speed with single-byte coding. For this reason, parallel multi-byte logic processing is often used in 8B/10B encoding to reduce the clock burden when the input data stream frequency is high. However, due to the constraints of the encoding logic, the encoding speed is significantly reduced when encoding multiple bytes in parallel [10-11], which makes the overall encoding effect unsatisfactory. This paper proposes an improved solution for the 8B/10B parallel encoder, which reduces the overall encoding delay by pipelining structure and optimizes the key delay paths, which can effectively improve the encoding speed while reducing the consumption of logic resources.

2. Parallel encoder structure analysis

8B/10B parallel coding circuits are traditionally constructed in both cascade and polarity-selective. Figure 1 shows a two-byte cascaded parallel encoder, and Figure 2 shows a two-byte polarity-selective parallel encoder. As Figure 1 shows, the output polarity of the first encoder is used as the input polarity of the second encoder, and the final output polarity is passed through a register as the input polarity of the next clock cycle of the encoder. This structure is low in resource usage and simple, but when the number of parallel bytes for 8B/10B coding increases, the coding delay increases, reducing circuit operation speed. Because the polarity logic is on the critical delay path and can only calculate the polarity value for the next level after the previous levels have been calculated, each additional input byte results in an additional level of delay.

The polarity-selective structure is proposed to solve this problem. As shown in Figure 2, for each encoder stage except the first, RD+ and R.D.- are encoded simultaneously, and the final polarity value with the 10B code is selected based on the output polarity of the previous encoder. The encoding process of the individual bytes in this structure is independent of each other so that when the input byte is increased, the encoder only increases the logic delay of the selector section.

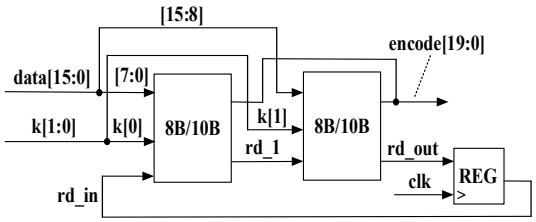

Figure 1. Two-byte cascaded parallel encoder.

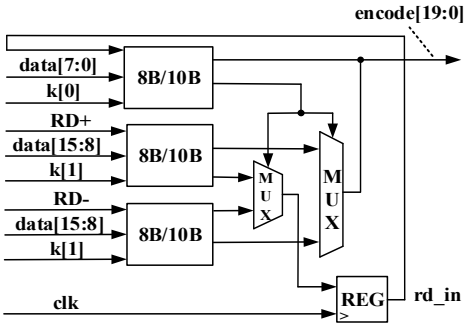

Figure 2. Two-byte polarity-selective parallel encoder.

Let the encoder encoding delay be T_e, where the encoding delay of the polarity operator logic is T_{rd}, and the selector delay is T_{mux}, table 1 shows the combination logic delay. Then the n-byte cascaded parallel encoder critical delay path delay is $T_e+(n-1)T_{rd}$, and the n-byte polarity selective parallel encoder critical delay path delay is $T_e+(n-1)T_{mux}$. Considering the process of encoding logic operations, it is clear that $T_{mux} < T_{rd}$, so the polarity selection structure reduces the delay of the polarity operation in the critical

delay path by replicating the operational logic. However, the selector delay of a multi-stage cascade will also gradually increase and become a key factor limiting the encoding speed. And the encoder adds a large amount of combinational logic circuitry every time the input bytes increase, which increases the resource usage of the encoder and reduces the stability of the structure. Therefore, the polarity-selective structure coding circuit still suffers from shortcomings.

Table 1. Combined logic delay

Combinational logic	Encoding logic	Polarity operator logic	MUX logic
Time taken	T_e	T_{rd}	T_{mux}

3. Novel 8B/10B parallel coding circuit design

Figure 3 gives a block diagram of this design's 8B/10B parallel encoding circuit. As shown in this block diagram, the overall encoding process is split into two levels of flow units when encoding multiple bytes of data in parallel. The first flow unit classifies the bytes according to the byte data and makes a polarity judgment based on the classification result. In this stage, the byte classification is carried out in parallel, while the polarity judgment is made by obtaining the polarity calculation of the previous polarity deviation value R.D. The byte classification and polarity judgment are completed in one clock cycle, and the R.D., k value, 8bit data, and 5bit data polarity judgment value P_5B are entered into the second stage of the flow cell via registers.

The second stage of the flow cell performs the pre-coding and coding correction. As the inputs required for pre-coding are already calculated in the previous stage, the pre-coding of the bytes can be carried out simultaneously. After pre-coding, the 10B code is corrected using the polarity obtained in the first stage flow cell, which is still in parallel. As a result, when the input bytes are increased, the encoding delay of the first-stage flowing unit increases, while the second-stage flowing unit is unaffected.

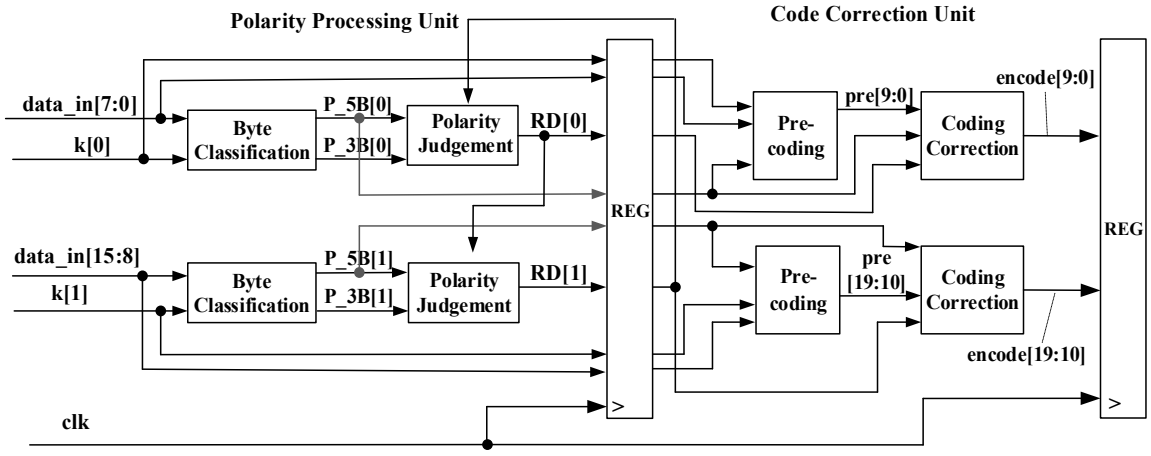

Figure 3. 8B/10B Parallel Encoding Circuit Structure for Polarity Pre-processing.

3.1 Design of the polarity processing unit

The polarity processing unit of the single-byte is shown in Figure 4, where 3bit data and 5bit data are classified according to code elements, but the classification process is not the same.

As shown in equations (1) to (4), "m" in L_{mn} represents the number of "1", and "n" represents the number of "0 ". The subscript "N" indicates the logical inverse of the value. The 5bit data is classified into four types according to the polarity classification requirements, and the 5bit data belonging to L_{22} is not identified.

$$L_{40} = A \bullet B \bullet C \bullet D \tag{1}$$

$$L_{40} = A_N \bullet B_N \bullet C_N \bullet D_N \tag{2}$$

$$L_{13} = (\mathrm{A} \oplus \mathrm{B}) \bullet C_N \bullet D_N + (\mathrm{C} \oplus \mathrm{D}) \bullet A_N \bullet B_N \tag{3}$$

$$L_{31} = (\mathrm{A} \oplus \mathrm{B}) \bullet \mathrm{C} \bullet \mathrm{D} + (\mathrm{C} \oplus \mathrm{D}) \bullet \mathrm{A} \bullet \mathrm{B} \tag{4}$$

The 6B and 4B codes of the 5bit data and 3bit data mapping have three polarities, where the 6B and 4B codes with polarities of +2 and -2 are called polar codes, and the 6B and 4B codes with polarities of 0 are called non-polar codes. The polarity classification identifies the 5bit data and 3bit data with polarity codes. When P_5B and P_3B are 1, it identifies the mapping of the data as polarity codes; when it is 0, it is non-polarity codes. The 5bit data classification result simplifies the operation process of 5B polarity classification. In comparison, there are only 23 combinations of 3bit data and fewer types of 4B codes mapped, so the polarity judgment is performed directly by skipping the byte classification process. The P_5B and P_3B operation logic is shown in (5)~(6).

$$P\_5B = L_{04} + L_{40} + L_{13} \bullet E_N + L_{13} \bullet D \bullet E + K + L_{13} \bullet E \tag{5}$$

$$P\_3B = F \bullet G \bullet H + F_N \bullet G_N \tag{6}$$

The part shown in dashed lines in Figure 4 is the polarity judgment module, which operates the logic as shown in equations (7) to (8). The result of the polarity judgment, i.e., the polarity deviation value R.D., is transmitted to the next encoder as the input polarity. This part of the operational logic is in the critical delay path. RD_in is the input polarity, and the default RD_in is 0 when this byte is the first byte of the input data stream. Similar to P_3B and P_5B, dp distinguishes 10B codes into polarity and non-polarity codes. When 4B and 6B codes have polarity in one and only one code group, the 10B code is polarised, and dp will be 1. If both code groups have polarity or neither polarity, the 10B code is unpolarised, and dp is 0.

$$dp = P\_3B \oplus P\_5B \tag{7}$$

$$RD\_out = RD\_\mathrm{in} \oplus dp \tag{8}$$

The gate-level circuit of the four-byte down polarity determination module is shown in Figure 5. When the input data stream is increased by one byte, the polarity coding code block is increased by two XOR gates, but only one XOR gate is cascaded. Assuming a delay of T_d for the combined logic of the byte classification module and T_{xor} for the individual XOR gates, the critical path delay of the first level of the flow unit is $T_d+(n+1)T_{xor}$ for an input data stream of n bytes. It can be seen that for each additional input byte, the delay of this level of the flow unit increases by only one XOR gate.

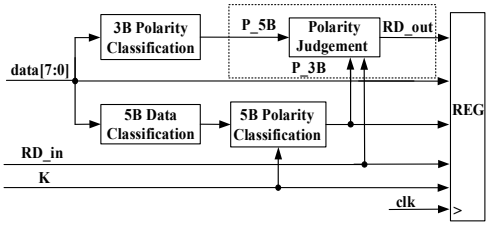

Figure 4. Polarity Processing Unit Block Diagram.

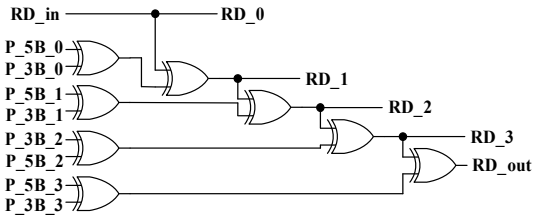

Figure 5. Gate-level circuit diagram of polarity judgment module.

3.2 Design of the code correction unit

The code correction unit also divides the 8bit data into 5bit data and 3bit data. The pre-coding and coding correction of the two parts is done in parallel, and the second stage flow unit operation of a single byte is shown in Figure 6.

When pre-coding 5bit data, previous data classification results can be used to simplify the operation. However, due to the large number of registers added by this method when passing between pipeline

levels, the traditional 5bit pre-coding method is used in this design, given the speed and area trade-off.

As the pre-coding values follow the minimum code element change principle in the code table, there is not only one case of the polarity of the pre-coded values and their required input polarity. Therefore, the 5bit and 3bit data need to be identified with the polarity of the pre-coding.

In the case of 3bit pre-coding, the polarity after 5bit encoding is required as a basis for judgment to eliminate the D.C. effect when 4B codes are combined with 6B. However, the polarity processing unit has already obtained the results of the polarity operation, so the 3bit pre-coding does not have to wait for the results of the 5bit pre-coding operation, and the two parts can be performed in parallel.

The coding correction module corrects the pre-coding values. The corrected polarity of the 6B code is the input polarity of the 10B code, i.e., the polarity deviation value R.D. The corrected polarity of the 4B code is the output polarity after the 6B coding. When the input polarity is unified with the pre-coding polarity mark, the module does not change the pre-coded result, indicating that the corrected coded value is the pre-coded value. When the input polarity is opposite to the pre-coded polarity mark, the correction code value is the inverse of the pre-coded value.

Since this part only adjusts the order of operations for the critical delay paths and does not change the logic operation, the logic formula for this unit is not repeated.

A theoretical analysis of the operating speed of the unit concerning the overall circuit will be shown. The combined logic delay of a single byte for this unit is assumed to be T_c, independent of the number of input bytes because the bytes in the encoding and correction unit operate in parallel. Thus, the minimum clock period throughout the encoder is $T = \max(T_d + (n+1)\,T_{xor}, T_c)$, and the maximum operating frequency is $\frac{1}{T}$ Hz. As the input bytes increase, the combined logic delay eventually results in $T_d + (n+1)\,T_{xor} > T_c$, where the data transfer rate is $\frac{10n}{T_d + (n+1)T_{xor}}$ b/s.

Therefore, a comparative analysis of this design with polarity-selective structures can be carried out. The analysis of the critical delay path gives the following table of gate usage for T_e and T_d, so it is clear to conclude that $T_e > T_d$. Although the critical delay path delay in T_d also requires the addition of two T_{xor} delays, this has no impact on the overall conclusion. As the number of input bytes increases, the advantage of this design does not change as long as T_{mux} does not fall far below T_{xor}.

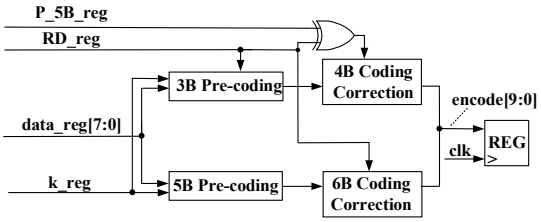

Figure 6. Structure diagram of code correction unit.

Table 2. The use of combinational logic in critical delay paths.

Combinational logic	Critical delay path	
	T_e	T_d
AND	5	3
OR	1	3
NOT	5	0
NOR	4	1

4. Circuit simulation and performance analysis

The four-byte parallel encoder was simulated using Synopsys' VCS software to prove the correct functionality of this encoder. The simulation results are shown in Figure 7 and Figure 8. Figure 7 shows the simulation results for the data code D code, and Figure 8 shows the simulation results for the control code K code. In the figure, data[31:0] is the input data, k[3:0] is the input K value, data_valid identifies the input data as valid, ser_data_0[9:0]~ser_data_3[9:0] is the coding result from byte 0 to byte 3, and ser_valid identifies the coding output as valid. The simulation results show that the design delays three clock cycles, the encoding result corresponds to the input data, and the multi-byte encoding function can be executed.

The SMIC 65nm process used Synopsys' Design Complier software to synthesize this design. Two 8B/10B parallel encoders were selected for comparison, using logic operations and aiming to improve coding performance. The literature [10] and [11] provide more representative recent designs, with [11] adapting the operational logic to accelerate the coding speed in a cascaded structure. The literature [10] further reduces the coding delay by splitting the coding and polarity selection of the polarity-selective structure into two levels of flowing units.

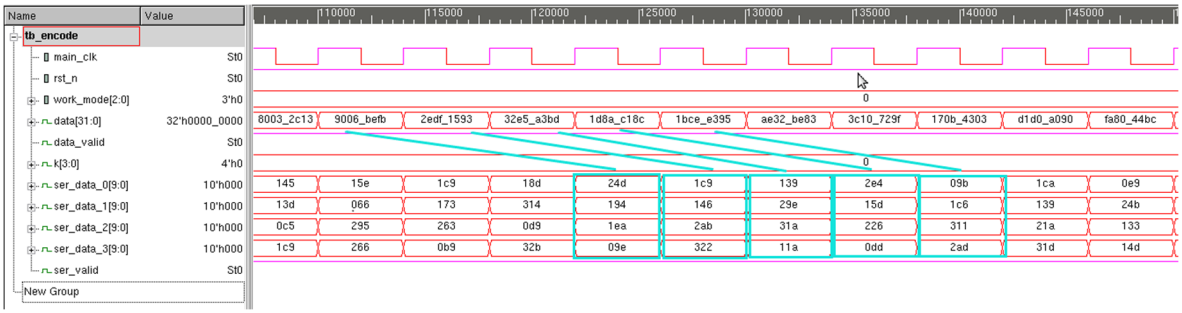

Figure 7. Data code D simulation results.

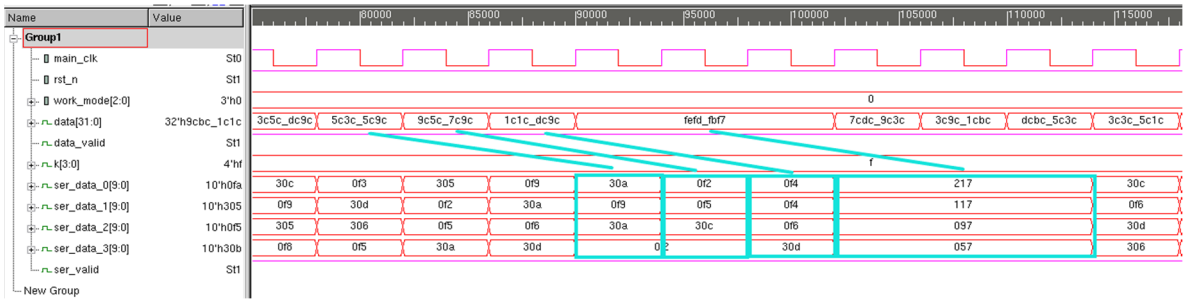

Figure 8. Control code K simulation results**.**

As can be seen from Table 2, the design runs at 38.2% higher frequency compared to the optimized cascade structure. This data demonstrates that the pipelined structure effectively reduces the overall coding latency by splitting the polarity pre-processing and coding into two levels of pipelined units. As the input bytes increase, the speed advantage of this design becomes even more significant due to the further increase in latency of the critical delay paths, and the degree of increase in area and power consumption associated with the pipelined structure is acceptable. Compared to the polarity-select structure, the maximum operating frequency of the design is similar to that of the polar select structure when both secondary pipelined structures are used. This design demonstrates that the optimization results for critical delay path delays are the same for both designs with four-byte inputs. The design also performs only one operation on each byte, thus consuming 82% of the logic resources and 83% of the power, a significant advantage over the polar select structure in resource usage.

Table 3. Comprehensive results and comparison of front-end results under SMIC65nm process.

	Novel Design		Literature [11]	Literature [10]
Lanes number	1	4	4	4
Maximum frequency (MHz)	>1000	909.10	657.89	877.19
Area (μm^2)	644.76	2245.68	1645.56	2714.76
Power (mW)	0.4146	1.3537	0.6410	1.6391

5. Conclusions

The encoding process is designed to improve the operation frequency of the 8B/10B parallel encoder. Using a pipelined structure that isolates the excessively long polarity operation delay path brought about by the operational logic under multi-byte input from the encoding operation, mitigating the impact of the long combined logic delay on the overall operation speed.

The 8B/10B logic operation is then modified based on the proposed parallel encoder structure by classifying the 8-bit data and performing a polarity operation. After which, the resulting polarity is used to directly encode the input bytes in parallel, in the process of which new logic operation formulas are obtained.

The critical path delay of this design is theoretically analyzed and compared. Synthesis demonstrates that this design can encode at a higher frequency while occupying a lower area and power consumption and has obvious application value.

Acknowledgments

This paper is one of the milestones of the cross-cutting project "Development of a memory laboratory and technical services regarding the Loewe project" (Z1500019001), thanks to the help of teachers and colleagues during the research process.

References

[1] Ashrafi, H., Mousazadeh, M., Hadidi, K. (2016) An 8B/10B encoder with 2GHz operating frequency. 2016 IEEE 59th International Midwest Symposium on Circuits and Systems. In: Abu Dhabi. pp. 1-4.
[2] WIDMER, A.X., FRANASZEK, P.A. (1983) A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code. IBM Journal of Research and Development, 27(5): 440-451.
[3] Cai, W.L., Zhao, J.Z., Lv, Y.J. (2019) Implementation and extension of 8B/10B encoder in PCIE2. Acta Scientiarum Naturalium Universitatis Nankaiensis, 52(2): 34-38.
[4] Wang, F., Zhou, L., Zhang, Z.F. (2016) Design and implement new structure 8B/10B encode. Microelectronics& Computer, 33(10): 151-154.
[5] Huo, X.H., Yao, Y.F., Jia, X.X., et al. (2015) Implementation of 8B/10B encoder based on JESD204B interface protocol. Chinese Journal of Electron Devices, 38(5): 1017-1021.
[6] He, J., Pu, J., Sun, W., et al. (2019) Design of 8B/10B encoder in JESD204B. Electronic World, 56(02): 110-112.
[7] Zhang, T., Tao, X.X., Liu, Y. (2019) A design of 8B10B encoding circuit in 8Gsps analog-to-digital converter[J]. Journal of Xi'an University of Posts and Telecommunications, 24(05): 47-52.
[8] Li, C.Q., Cheng, J., Li, L., et al. (2017) Design of JESD204B transmitter interface using parallel 8b/10b encoder. Microelectronics & Computer, 34(8): 70-85.
[9] Shu, Z.X., Huang, L., Du, X.L. (2015) A new kind of 8B/10B encoding design. Microelectronics & Computer, 32(9): 181-184.
[10] Chang, H., Ke, D.M., Meng, J., et al. (2018) Design and implementation of novel 8B/10B encoding. Computer Engineering and Applications, 54(2): 87-90.
[11] Wang, J.J., Wan, S.Q., Ji, H.C., et al. (2020) Design and implementation of 8B/10B parallel coding circuit for JESD204B protocol. Microelectronics & Computer, 37(6): 35-39.