

Several Efficient Alternative Solutions for RS Codes in RAID 6

Xinran Xue *

College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen, 518060, China

* Corresponding Author Email: 2021270040@email.szu.edu.cn

Abstract. The RAID-6 specification requires a storage system with multiple storage devices to be able to endure the failure of any two parities. RAID-6 can be implemented using a variety of erasure coding methods, but each has drawbacks, like low fault tolerance, bad writing performance low decoding speed, and so on. In this paper, we research a variety of materials, explore RAID 6 disks to comprehend their purpose and prerequisites, and understand how to decode RS codes. In order to help people find a decoding method that is more appropriate to their own systems and actual needs, we research and compare numerous high-quality alternating decoding techniques, and the RAID 6 Liberation codes and Row-Diagonal Parity emerge as the winners. We introduce their decoding methods briefly and list their advantages and disadvantages in this paper for people to request based on their needs. Finally, we compare different decoding methods, propose practical application scenarios and future development directions for these more efficient decoding methods, and also point out their existing shortcomings.

Keywords: RS Codes, RAID 6, the RAID 6 Liberation Codes, Row-Diagonal Parity.

1. Introduction

As storage systems continue to grow in size and complexity, the adoption of RAID-6 fault tolerance has become increasingly widespread. RAID-6 [1] is a defined standard for storage configurations comprised of multiple storage devices, designed to withstand the failure of two devices concurrently. In recent years, due to the increasing number of simultaneous failures of blocks on one disk drive and another disk, RAID 6 has gained a more and more important role. RAID 6 is a regulation designed for a storage system containing $k+2$ storage nodes, which is able to tolerate up to 2 failure disks. Reed-Solomon codes (RS codes) are a type of error-correcting code used in data communication and storage systems like RAID 6. These codes are widely employed in applications where data must be transmitted or stored reliably in the presence of errors or data corruption. RS codes are capable of correcting multiple errors within a block of data. The number of errors that can be corrected is determined by the code's parameters.

RAID 6 is a data storage method that belongs to the RAID family, which stands for Redundant Array of Independent Disks. In contrast to various other RAID configurations, RAID-6 isn't a precise methodology but rather a defined standard. It can withstand the failure of any two disks [2], which requires erasure codes. Although there have been lots of erasure codes, each has its own drawbacks and no one can be a whole-winner. Thus, many studies about erasure codes like Liberation Codes and Row-Diagonal Parity are currently improving these algorithms [3]. RAID 6 also has some relative research in reality. The hardware development of NAND flash memory controller that can be used for the online pipeline inspection process of pipeline detectors, using RAID-6 to protect the information integrity scheme [4].

RS codes have a useful algebraic framework that is capable of correcting not just common errors but also unforeseen error patterns [5,6]. The growth of the positioning, navigation, and timing (PNT) market demanding the service related to GNSS signals asks for much-improved accuracy [7,8], and the CSK modulation combined with Reed-Solomon (RS) codes helps it achieve the increased data rate. It implements a low-complex iterative symbol-level decoding scheme for Reed-Solomon codes but only works at a relatively lower complexity [9]. RS codes offer significant advantages in enhancing RAID performance, and it remains to be seen whether they will find additional practical applications in the future.

Array codes, such as RAID, are widely used in data storage systems to improve data reliability [10]. But nowadays, as people's demand for data storage is growing, though the capacity of hard disks is also increasing, the fault tolerance rate has little improvement. It can be seen that the reliability problem needs to be solved urgently.

In this paper, we will first briefly introduce the application and performance analysis of RS code in RAID 6. We then propose two excellent alternatives and describe their specific work mode in RAID 6, and point out their strength and weaknesses. Finally, the three algorithms are compared, and the prospects and suggestions for the future are put forward.

2. Reed-Solomon Coding for RAID-6

2.1. Reed-Solomon Coding

Reed-Solomon (RS) codes, sometimes referred to as Reed-Solomon error correction codes, are a type of error-correcting code widely used in data storage and transmission systems. Reed Solomon codes were first proposed by American engineers Irving S. Reed and Gustave Solomon in the early 1960s [1]. Their invention is considered a major breakthrough in coding theory, as they provide an efficient and reliable error correction coding method that enables data to be stored and transmitted under harsh conditions.

RS code is a special algorithm that divides the data on each device into w -bits words and then lets the i -th byte on the device add redundancy uses this special algorithm ($GF(2^w)$) to calculate a corresponding value, which makes use of the value in the previous devices [1]. This algorithm is similar to the XOR operation, but it makes use of multiplication, so the amount of computation is larger for the machine, and the cost of application operation in RAID becomes more expensive. This coding technique was protected as patents in the early stages when it first came out, and companies that purchased its patents could use it for storage system upgrade protection and gain profit from it, so it was popular with companies at that time. Their widespread application in data storage systems is due to their reliability and efficiency, as well as their powerful performance in data integrity.

2.2. Reed-Solomon Coding for RAID-6

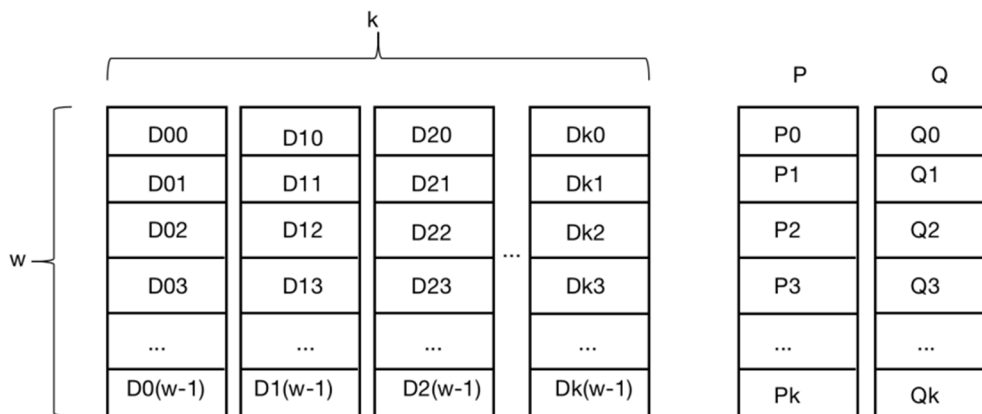


Figure 1. Reed-Solomon Coding for RAID-6

(Photo/Picture credit: Original)

Figure 1 briefly shows the application of RS code in RAID 6. D is the original storage node, and P and Q are two data nodes used to add redundancy which should be calculated using a selected method. As a whole, RAID 6 specifies a $k+2$ storage system, where each node contains B bytes and is divided into W -bit words. The storage capacity of each node is kB , and the remaining two disks (P and Q) use $2B$ of storage capacity. P and Q are also called p parity and Q parity, and their values are calculated based on the previous k basic disks. The value in P is simply the sum of the values in the four squares of the same row in the previous four basic disks. The value of P is a special superposition of the values of the same row in the 4 base disks using the RS code algorithm. In this way, when up

to 2 disks fail, their data values can be recovered from the remaining disks. Not to mention, when p and Q are destroyed, it is possible to quickly calculate the alignment value from the first four base disks as well.

However, this kind of coding method also has a disadvantage in that, as described for RS code in 2.1, the algorithm of RS code requires multiplication, which requires a very large amount of computation in the data storage system, which will reduce the speed of encoding. Therefore, there are many articles to study and improve this problem. Recently, a new algorithm called Cauchy Reed-Solomon Coding has been proposed. In the case of w is a power of 2, the multiplication of 2 can be calculated quickly, so the encoding performance can be greatly improved. However, it should also be noted that the modification and decoding performance have no effect, so since there is only the encoding speed improved, still a large room for further research in the decoding and modification performance to be got through.

2.3. Performance of Reed-Solomon Coding for RAID 6

RS codes have excellent error correction ability, also called decoding performance, which can detect and correct errors or failures in data. This means that even if some errors or losses occur during data transmission or storage, RS codes can still recover the original data, ensuring the integrity and reliability of the data. RS codes was given patent protection early on, which also contributed to its appeal as a technology in the past because holders of patents could manage its use and possibly profit from it. As a result, RS codes are frequently utilized nowadays. However, RS codes is often slow, and the performance of its new technology for modifications is only average. Alternative encoding methods that come close to ideal performance are therefore crucial.

3. The RAID-6 Liberation Codes

3.1. Liberation Codes

In order to solve the problems of poor encoding performance and average modification performance of RS codes, we find a relatively superior encoding method called Liberation Codes. Figure 2 briefly introduces its bit matrix encoding method.

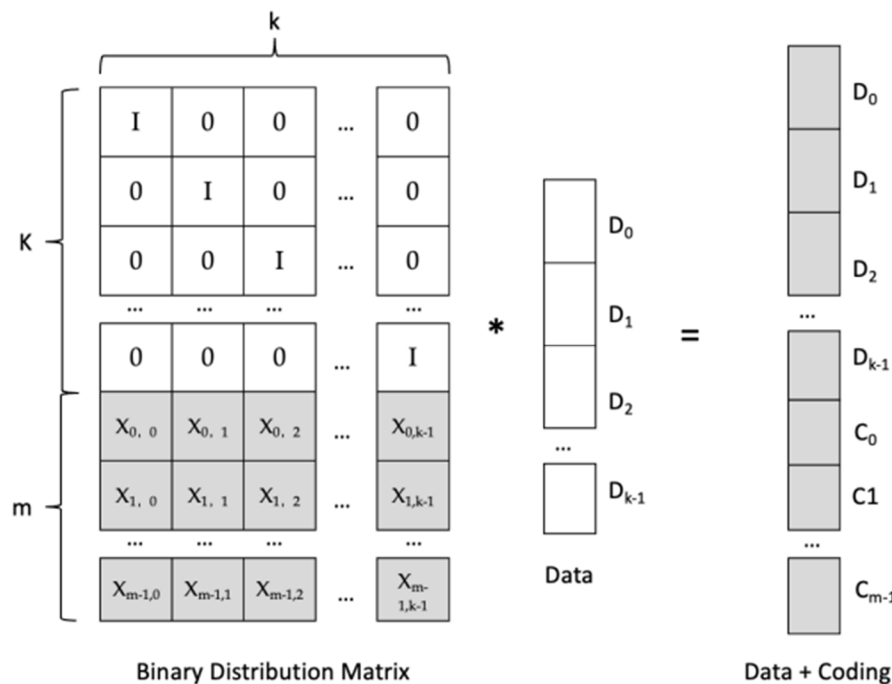


Figure 2. Liberation Codes Encoding Calculation Method
 (Photo/Picture credit: Original)

D represents raw disk data, with a total of k disks and each disk contains w elements. While encoding, we need to first create a binary distribution matrix, abbreviated as BDM. The upper half of the matrix consists of a k-order identity matrix, where k corresponds to the number of disks k. The lower half of the matrix is an m*k-sized artificial matrix X, with specific creation specifications detailed in many papers dedicated to Liberation Code [1].

Each disk corresponds to a row in the Binary Distribution Matrix and a row in ‘Data + Coding’. When decoding, we assume that we have x surviving disks holding the original data, and we can use these surviving data to create the corresponding BMD matrix. The new BDM matrix can be multiplied by the original data to obtain these x surviving data. Therefore, our raw data can be obtained by multiplying the inverse of the BDM matrix with x surviving data, thereby decoding to obtain the lost data.

3.2. Liberation Codes for RAID 6

Due to the direct correlation between the performance of encoding and the number of encodings [1], when applying Liberation Codes to RAID 6-bit matrix encoding, we will minimize the number of encodings to improve encoding speed and performance. The specific operation is shown in figure 3: define X_i yourself and the specific standardization methods have been elaborated on in many articles [1, 2, 11], so we won't go into detail in this paper. One noteworthy point is that the value of w is limited and depends on the value of k, and it must be a prime number greater than or equal to the value of k. Liberation Codes specifications minimize quantity but maintain MDS. In this way, we can achieve both fast encoding and better decoding performance. The specific comparison with other methods will be presented in the next chapter 5.

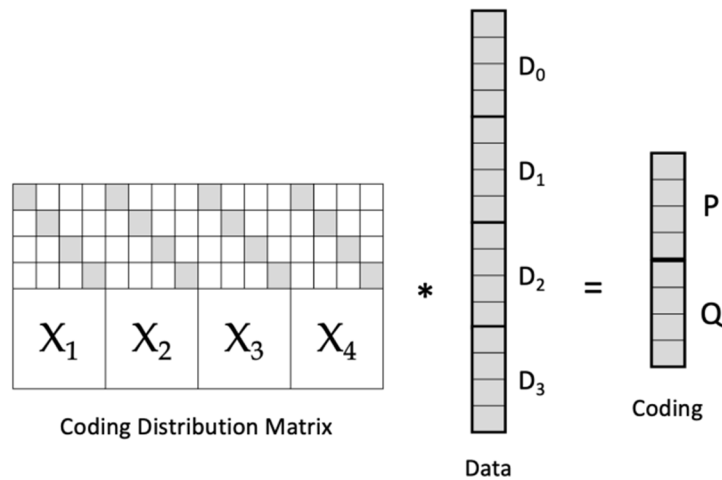


Figure 3. Liberation Codes Encoding Calculation Method
 (Photo/Picture credit: Original)

3.3. Performance of the RAID-6 Liberation Codes

Better Encoding Performance. We use the average number of XOR operations required per codeword to judge the performance of the encoding. Many studies have shown that when the big value of w is fixed, fewer XORs are needed for Liberation Codes, which leads to higher speed while encoding and better encoding performance. But we also need to note that this is at the expense of greater memory consumption. Therefore, it is advisable to choose the appropriate encoding method that fits our need according to the actual situation of our system capacity and hard disk performance.

Higher Compatibility for Different Disk Sizes. Besides, it has higher compatibility for different disk sizes. Figure 1 simply presents the RAID 6 system. To build flexible RAID-6 systems, it is often it is advisable to allow k to grow and shrink dynamically within limits. However, as they are unable to adjust by reducing was well, other approaches, such as RS coding, perform worse when k decreases.

However, it will not be an issue with RAID-6 Liberation Codes, making it more flexible. Therefore, the greater compatibility gives it superior performance.

The Benefits and Disadvantages of Bit Scheduling. Finally, it enjoys faster decoding speed thanks to a more advanced decoding algorithm, enabling it to decode at a rate between ten and fifteen percent over optimal, which is much faster than RS code.

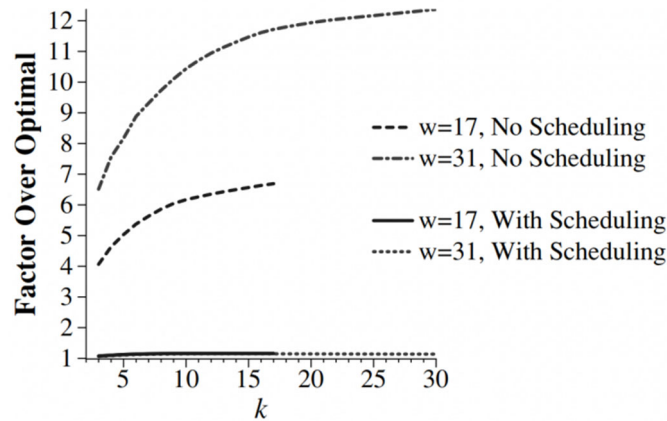


Figure 4. The Effectiveness of Bit Matrix Scheduling on Liberation Decoding [1]

However, it achieves so due to an algorithm requiring bit scheduling. In Figure 4, The vertical axis represents the number of XORs per failed word and shows the average value, so when the elements of each disk are equal, which is 17 and 31 respectively, the number of decoding failures is close to zero with scheduling, But without it, the error rate increases dramatically, and the number of xor operation failures in $w = 17$ and $w = 31$ increases from 4 and 6 to 6 and 12, respectively, as the number of disks k increases from 5 to 30. Thus, Liberation codes would be unusable as a RAID-6 technique anymore. It remains a topic of future work to see if the scheduling algorithm may be improved further. Thus, there is still much future work that could be done in this field.

Also, compared with some long-history coding methods like RS codes, the RAID-6 liberation code may have less mature implementations available. In contrast, it has less knowledge support and less persuasiveness and authority as it has not yet been widely applied.

4. Row-Diagonal Parity

4.1. Row-Diagonal Parity

Row-diagonal parity is a technique used in error detection and correction in computer memory systems, particularly in systems with memory modules arranged in a two-dimensional grid or matrix. It is a method to detect and sometimes correct errors in data stored in such memory modules.

The raw data in the disks is organized into rows and columns, creating a two-dimensional matrix of memory cells. Each cell stores a binary value (0 or 1). In addition to the data rows, there are parity rows added to the matrix. These parity rows contain parity bits, which are calculated based on the values in the corresponding data rows. Each data row has its row parity bit. This row parity bit is calculated by counting the number of 1s in the data row. If the count is even, the row parity bit is set to 0; if it's odd, the row parity bit is set to 1. Diagonal parity bits are calculated by taking a diagonal slice of the memory matrix (from the upper left to the lower right or vice versa). The diagonal parity bit is calculated by counting the number of 1s in this diagonal slice. Again, if the count is even, the diagonal parity bit is set to 0; if it's odd, the diagonal parity bit is set to 1.

During read operations, the row and diagonal parity bits are recalculated based on the data read from memory. If there's an error in any of the data bits, the parity calculations will not match the stored parity bits, indicating an error. This allows for the detection of errors in the data.

4.2. Row-Diagonal Parity for RAID 6

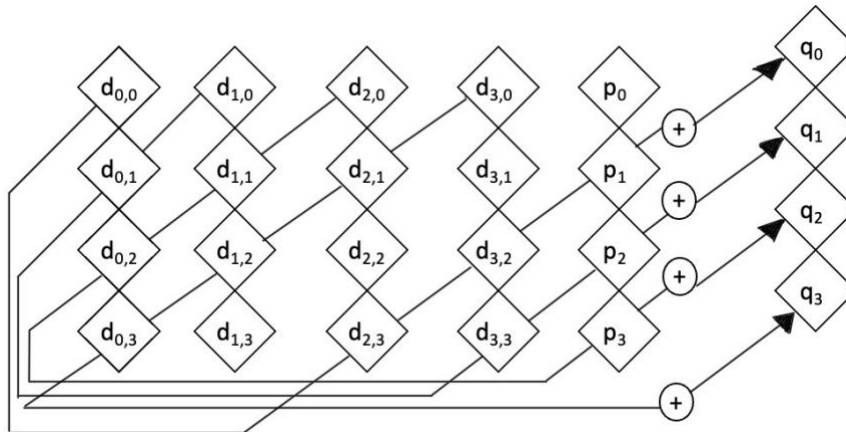


Figure 5. RDP Coding with $k = 4$ and $w = 4$

(Photo/Picture credit: Original)

Figure 5 is a simple example of a stripe applied in RAID 6 using Row-Diagonal Parity (RDP). A stripe has a total of $p+1$ disks, where p is a prime number greater than 2. The p in the example is 5, thus consisting of a total of 6 disks, 4 raw disks, 1 p parity, and 1 q parity. $d_{i,j}$ represents a disk that stores raw data, with each column being a disk, where each disk is divided into m regions. In this example, set m as 4 in the figure, and this disk is divided into 4 regions. P and Q are parities, used to increase redundancy. The value of p is the simple sum of the values in k regions on the same row in the original disk, like $p_0 = d_{0,0} + d_{1,0} + d_{2,0} + d_{3,0}$. The value of Q is the sum of the values of the regions connected by diagonals. The connection method has been expressed very concisely in Figure 5, such as $q_0 = d_{0,0} + d_{2,3} + d_{3,2} + p_0$. One of the diagonals is not connected, and the values on this diagonal are not stored or calculated, so it is also known as a missing diagonal [10].

This encoding method has been proven by many articles to apply to RAID 6 and has excellent performance. Its construction process has just been explained and is very simple, so RDP has been proven to be optimal in terms of computational complexity during the construction and reconstruction process. At the same time, it also has excellent performance in storing and accessing data.

4.3. Performance of Row-Diagonal Parity

Write performance in RDP can be slower compared to RAID levels that only use striping (e.g., RAID 0) because it involves not only writing the data but also calculating and writing the parity blocks for both rows and diagonals. This additional parity calculation can introduce some write overhead, but it also provides RDP with decoding ability. Therefore, since algorithms such as RDP can provide RAID with better decoding performance, we should compare and analyze RDP with other algorithms that applied to RAID 6 when analyzing its performance.

Encoding Performance: Since the computation method of RDP is based on XOR operations, if there are more operations required, it may need more time and a better-performing device to encode. In other words, if more XOR operations are required, the cost of computation will be more expensive.

Table 1. Per Row XOR Counts for Parity Construction

Data disks	RDP	EVENODD	Difference
4	6	6.67	11.1%
6	4	10.8	8.0%
8	14	14.86	6.1%
12	22	22.91	4.1%
16	30	30.93	3.1%

From Table 1, we can easily see that according to different rows, when the number of disks is equal, the number of XOR operations required for RDP is clearly less than EVENODD, a coding

method that is very similar to RS codes. Constructing diagonal parity involves performing the same number of XOR operations as constructing row parity, which is equal to $(p - 1)(p - 2) = p^2 - 3p + 2$. When considering both row and diagonal parity construction, the total number of XOR operations required amounts to $2p^2 - 6p + 4$ [12]. Thus, it can encode at a higher speed, leading to better writing performance due to fewer XOR operations, especially in the case with random writes.

Decoding Performance: RDP provides a high level of fault tolerance, allowing for multiple drive failures to be recovered. The computational expense of recovering data in the event of a failure within the Row-Diagonal Parity (RDP) scheme is also highly efficient. To restore data from a single disk failure, precisely $(p-1)(p-2) = p^2 - 3p + 2$ XOR operations are necessary. This is because each of the $p - 1$ missing row or diagonal parity sets consists of p blocks of the same size, and the lost block in each set is reconstructed by performing $p-2$ XOR operations on the remaining $p - 1$ blocks.

Table 2. Per Row XOR Counts for Data Reconstruction

Data disks	RDP	EVENODD	Difference
4	6	9.67	61.2%
6	10	13.80	83.0%
8	14	17.86	27.6%
12	22	25.91	17.8%
16	30	33.93	13.1%

Table 2 shows that RDP requires fewer XOR operations compared with other methods, so it has better coding Performance as well.

5. Application scenario

Data Storage in Pipeline Tester: RS codes, RDP, and Liberation Codes are all algorithms that could be used in the RAID 6 storage system. RAID 6 can be applied to the redundant data storage system of the online pipeline tester [3]. Because the pipeline is very long, the data obtained from each inspection needs a system with good storage capacity, so RAID is a very good choice. When corresponding to the storage algorithms needed by different pipelines, choose the most appropriate one according to the needs.

RS codes offer adjustable error correction levels, providing varying degrees of data redundancy and protection as needed. This makes them highly valuable in pipeline data inspection and storage. RDP primarily focuses on providing data redundancy for local storage and is not directly applied to this situation. RAID 6 Liberation Codes offer a high level of disk redundancy, ensuring data integrity even in the face of multiple disk failures. However, since its construction method is more complex compared with others, this level of redundancy may be considered overly complex and costly for pipeline data inspection.

Network Communications: These methods used in RAID 6 can help recover missing or corrupted data, making them valuable for applications such as wireless communication, satellite communication, and data transmission over noisy channels.

RS codes are commonly used for error correction in network packet transmission. They offer excellent error correction capabilities, capable of detecting and repairing lost or corrupted data blocks. The error-correction capability of RS codes can be adjusted based on the configuration to accommodate varying levels of noise and data loss. RDP is typically employed in RAID storage systems rather than directly in network communications. It provides error correction at the data block level, allowing for the recovery of damaged or lost disk data. However, in network communications, RDP is not mature enough to be applied in reality. RAID 6 Liberation Codes are specifically designed for RAID 6 and are used to protect data in storage arrays from the impact of multiple disk failures. In network communications, it is not typically the preferred error correction method due to its complexity, which may result in higher computational overhead.

6. Conclusion

In this paper, we introduce the principles of RS codes, RDP and Liberation Codes, as well as the specific methods of their application in RAID 6. We also analyzed their respective strengths and weaknesses. RS codes dominate in a wide range of application scenarios, but their speed is relatively slow. Liberation code has better decoding performance and higher compatibility with various disks, but requires bit scheduling and is more complex. The RDP encoding is simple and fast during construction, and the decoding performance is also excellent, but the limitations of the applicable scenarios are very strong.

Analyzing and comparing common and efficient encoding methods provides us with more intuitive and efficient guidance for future decisions regarding the use of different encoding systems for RAID 6. While making a choice between different parity systems, we should consider our specific storage requirements.

In future work, there is still significant room for improvement in these three encoding methods. For example, RS codes need to enhance their speed, Liberation Codes require complexity reduction and modulation improvements, and RDP needs further exploration of its application scope.

References

- [1] Qiyun, Cheng.: Short-Term Load Forecasting Model and Method for Power System Based on Complementation of Neural Network and Fuzzy Logic. Transactions of China Electrotechnical Society (2004).
- [2] Na, B., Zhang, Y., Liu L. and Liu P.: TSHOVER: A novel coding scheme for tolerating triple disk failures in RAID/DRAID. Tsinghua Science and Technology, vol. 12, no. S1, pp. 39 - 44 (2007).
- [3] Z, Huang., H, Jiang., Z, Shen., H. Che., N. Xiao. and N. Li., Optimal Encoding and Decoding Algorithms for the RAID-6 Liberation Codes. 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 708 - 717 (2020).
- [4] Rodríguez-Olivares, N. A., Gómez-Hernández, A., Nava-Balanzar, L., Jiménez-Hernández, H. and Soto-Cajiga, J. A., FPGA-Based Data Storage System on NAND Flash Memory in RAID 6 Architecture for In-Line Pipeline Inspection Gauges. IEEE Transactions on Computers, vol. 67, no. 7, pp. 1046 - 1053 (2018).
- [5] Xiao, Y., Luo, C. and Yang, C., The comparative analysis of LDPC and RS code. 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 4510 - 4513 (2011).
- [6] Cho, H., Song, H Y., Ahn, J M., et al.: Some new RS-coded orthogonal modulation schemes for future GNSS [J]. ICT Express (2021).
- [7] X, Zhang.: VLSI Architectures for Reed–Solomon Codes: Classic, Nested, Coupled, and Beyond. IEEE Open Journal of Circuits and Systems, vol. 1, pp. 157 - 169 (2020).
- [8] Hyunwoo, Cho., Hong-Yeop, Song., Jae Min, Ahn., Deok Won, Lim.: Some new RS-coded orthogonal modulation schemes for future GNSS, ICT Express, Volume 7, Issue 4, Pages 530 - 534 (2021).
- [9] Achala, G., Acharya, U. S. and Srihari, P.: On the Design of SSRS and RS Codes for Enhancing the Integrity of Information Storage in NAND Flash Memories. IEEE Access, vol. 11, pp. 73198 - 73217 (2023).
- [10] Hou, H., Han, Y. S., Shum, K. W. and Li, H.: A Unified Form of EVENODD and RDP Codes and Their Efficient Decoding. IEEE Transactions on Communications, vol. 66, no. 11, pp. 5053 - 5066 (2018).
- [11] Gang, W., Xiaoguang, L., Sheng, L., Guangjun, X. and Jing, L.: Constructing Liberation Codes Using Latin Squares. 14th IEEE Pacific Rim International Symposium on Dependable Computing, pp. 73 - 80 (2008).
- [12] Corbett, P., English, B., Goel, A., et al.: Row-diagonal parity for double disk failure correction [J]. USENIX Association (2004).