International Journal on Recent Researches in Science, Engineering & Technology (IJRRSET)

A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. Journal Indexed in JIR, DIIF and SJIF.

Available online at: www.jrrset.com

Received on: 04.12.2019 Published on: 21.01.2020 Pages : 51-57

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 7, Issue 12
December 2019.
JIR IF : 2.54
SJIF IF : 4.334
Cosmos: 5.395

ENCODING AND DECODING TECHNIQUES USED IN THE PHYSICAL CODING SUBLAYER USING THE UNIVERSAL VERIFICATION METHODOLOGY.

Dr. ROSI. A, Dr VENKATESH BABU, SANIYA SALIM, SHILPA ALIYAS,
Department of Electronics and Communication Engineering, INDIRA GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY NELLIKUZHI P.O, KOTHAMANGALAM, ERNAKULAM (DIST) PINCODE 686691

Abstract

Physical coding refers to the process of encoding information into a physical form, such as electrical signals or light pulses, for transmission or storage purposes. Sub-layer is used in radio communication systems to provide high-speed data transfer and prevent imbalanced data flow. PCS is a standard component of both 10Gb/s networks and it fulfils all the necessary functions of XGMII (10Gigabit Media Independent Interface). These functions include encoding and decoding XGMII data bits, synchronising code-groups, and supporting the IEEE 802.3 Media Access Control (MAC). This study primarily focuses on the mechanisms of encoding and decoding used in PCS. On the transmitter side, we will address the methods of data encoding and the specific guidelines for encoding the data. Similarly, on the receiving end, it is important to understand the process of decoding the data and the specific guidelines that must be adhered to in order to successfully decode the data. The PCS is emulated in the Verilog compiler simulator (VCS) using the Universal Verification Methodology (UVM), and the findings are shown in this document.

Keywords: Encoding and Decoding, Physical Coding Sub-layer, Common Public Radio Interface, UVM

1. Introduction

The radio base station system has two essential subsystems: the Radio Equipment Control (REC) and the Radio Equipment (RE). The Common Public Radio Interface (CPRI) is a communication interface inside a radio system that links two base stations, REC and RE [1]. This interface has many levels, with particular emphasis on the two bottom layers: the Physical Layer (layer 1) and the data connection layer (layer 2). The Physical layer is crucial in the CRPI interface. The Physical layer has a sublayer called the Physical Coding Sublayer (PCS). The main goal of this research is to expedite the transfer of data and address any discrepancies in code flow by using the Physical Coding Sublayer (PCS).

2. Physical Coding Sub Layer

The PCS and Gigabit Media Independent Interface (GMII) establish communication over 8-bit parallel data channels and many control lines. The PCS is tasked with converting each octet received from the GMII into ten-bit code groups. The PCS is tasked with converting ten-bit code groups received from the PMA into octets, which are then used by the top layers.

2.1 PCS Transmitter

The PCS transmitter block diagram, shown in Figure 1, includes an Encoder that consists of Encoding schemes and Mux. The encoding strategy is executed based on the selected lines. If the choose line is specified,

00: The system will use the 8b/10b encoding mechanism.
01: The system will utilise the 16b/20b encoding mechanism.
10: The system will utilise the 32b/40b encoding mechanism.
11: The system will operate using the default settings.

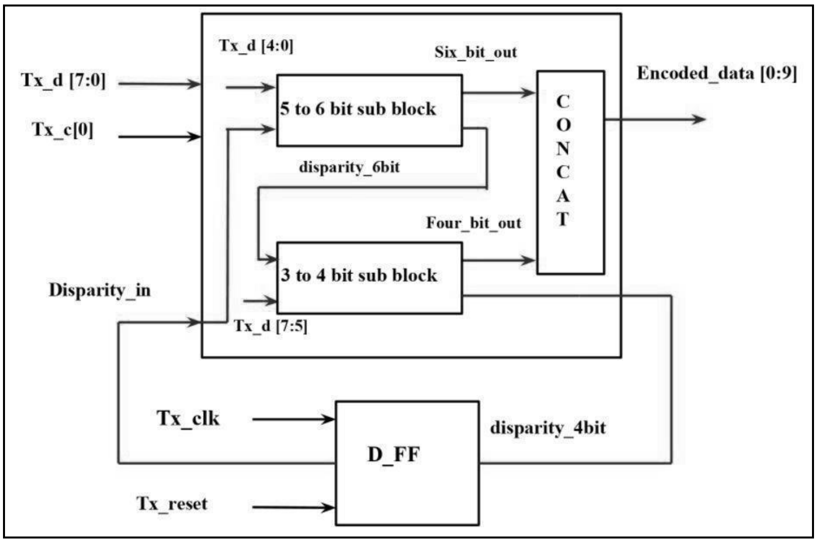


Figure 1 PCS Transmitter Blok Diagram

Disparity Rules:

- The idea of dispersion is used to achieve a balanced amount of 0s and 1s in order to establish a DC-balanced data stream.
- The disparity of a block is determined by subtracting the count of 0s from the count of 1s.
- The running disparity at the conclusion of each sub-block is positive when it sends a greater number of ones.
- The running disparity at the conclusion of each sub-block is negative when it sends a greater amount of zeros.
- The running disparity is considered neutral when an equal number of ones and zeros are sent.

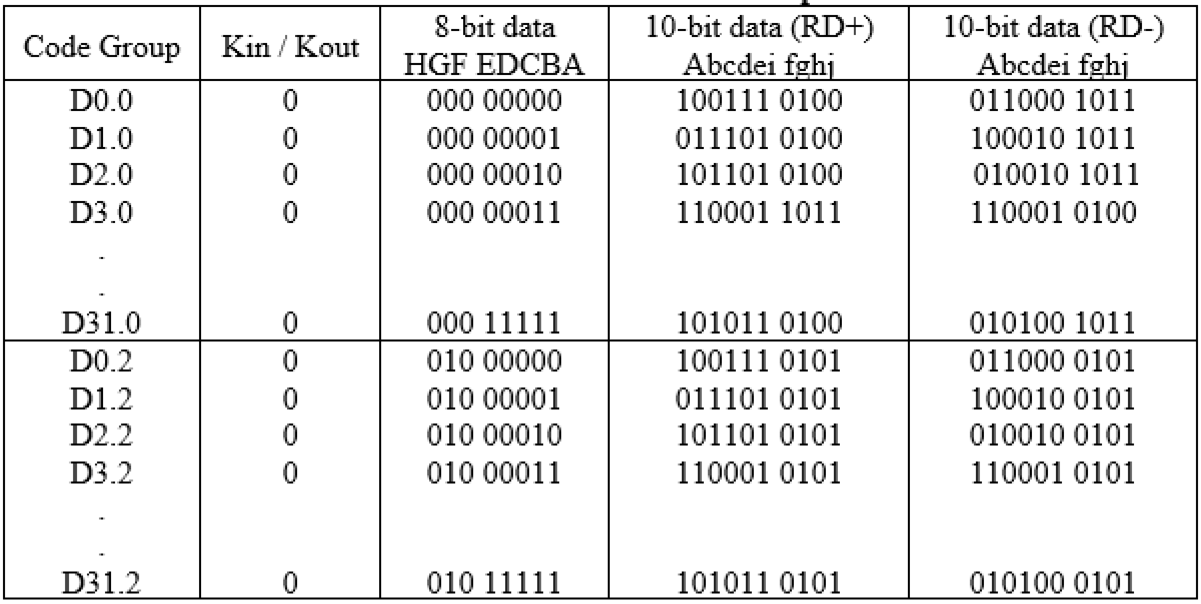
Data Code Groups and Control Code Groups: Table 1 represents Valid Data Groups and is denoted as DX.Y, where x runs from 0 to 31 and Y ranges from 0 to 7.The input consists of any of the 256 possible 8-bit combinations of DX.Y. The output that is delivered relies on the value of the Running Disparity. Table 2 represents the Control Data Groups as KX.Y. There are a total of 12 special control characters. Each 8-bit combination in the KX.Y table produces a valid output based on the Running Disparity. The operational mechanism of the Transmitter Block schematic is The input data, consisting of 8 bits, is separated into two sub-blocks: one with 5 bits and the other with 3 bits. The initial 5-bit is transformed into a 6-bit format according to the specifications of the Valid Data Group and Control Group. The input to the 3-bit block is the disparity of the 6-bit block, which is then turned into a 4-bit block. The result of this conversion, known as the running disparity, is then used as input for the 5-bit block. The final output consists of the concatenation of a 5-bit block and a 3-bit block, resulting in a 10-bit output.

8b to 10b Encoding is a process that converts incoming 8-bit data into 10-bit data using acceptable code groups according to the IEEE 802.3 standard.

16b to 20b encoding process involves converting a 16-bit input data into a 20-bit data by using two 8b to 10b blocks.

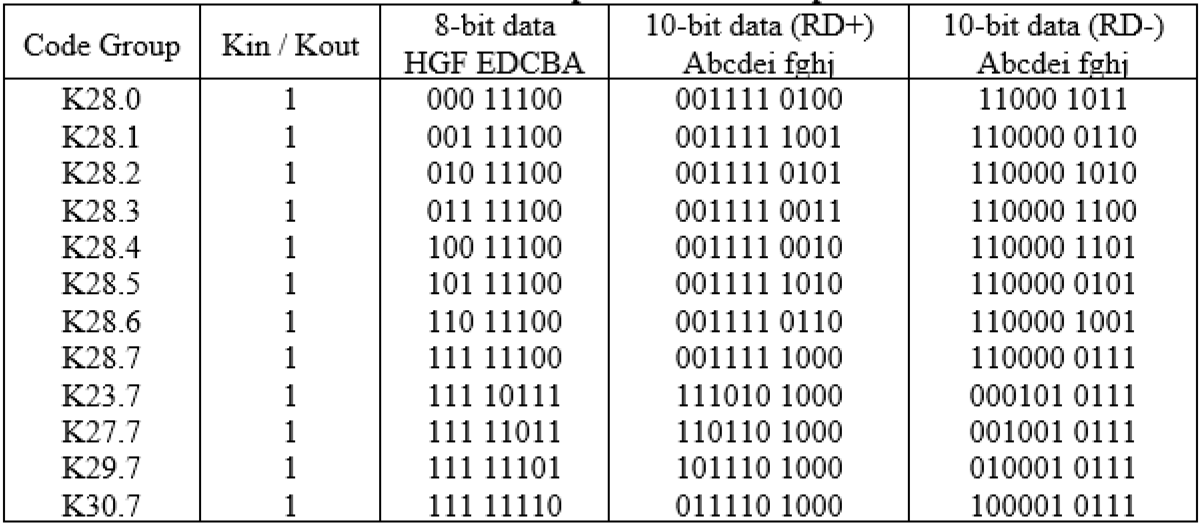
32b to 40b encoding process involves converting a 32-bit input data into 40-bit data by using four 8b to 10b blocks.

Table 1 Some Valid Code Groups



Code Group	Kin / Kout	8-bit data HGF EDCBA	10-bit data (RD+) Abcdei fghj	10-bit data (RD-) Abcdei fghj
D0.0	0	000 00000	100111 0100	011000 1011
D1.0	0	000 00001	011101 0100	100010 1011
D2.0	0	000 00010	101101 0100	010010 1011
D3.0	0	000 00011	110001 1011	110001 0100
.				
.				
D31.0	0	000 11111	101011 0100	010100 1011
D0.2	0	010 00000	100111 0101	011000 0101
D1.2	0	010 00001	011101 0101	100010 0101
D2.2	0	010 00010	101101 0101	010010 0101
D3.2	0	010 00011	110001 0101	110001 0101
.				
.				
D31.2	0	010 11111	101011 0101	010100 0101

Table 2 Valid Special Code Groups



Code Group	Kin / Kout	8-bit data HGF EDCBA	10-bit data (RD+) Abcdei fghj	10-bit data (RD-) Abcdei fghj
K28.0	1	000 11100	001111 0100	11000 1011
K28.1	1	001 11100	001111 1001	110000 0110
K28.2	1	010 11100	001111 0101	110000 1010
K28.3	1	011 11100	001111 0011	110000 1100
K28.4	1	100 11100	001111 0010	110000 1101
K28.5	1	101 11100	001111 1010	110000 0101
K28.6	1	110 11100	001111 0110	110000 1001
K28.7	1	111 11100	001111 1000	110000 0111
K23.7	1	111 10111	111010 1000	000101 0111
K27.7	1	111 11011	110110 1000	001001 0111
K29.7	1	111 11101	101110 1000	010001 0111
K30.7	1	111 11110	011110 1000	100001 0111

2.2 PCS receiver

The receiver employs a decoding method to encode the data received from the transmitter, as seen in Figure 2. The decoder's input consists of a 10-bit binary number that is divided into a 6-bit portion and a 4-bit portion. The most significant 6 bits are sent as input to a decoding table with 6 bits, while the least significant 4 bits are provided to a decoding table with 4 bits. The input to the 4-bit block is the running disparity of the 6-bit most significant bit (MSB), whereas the input to the 6-bit block is the disparity of the 4-bit least significant bit (LSB). Any 10-bit data from either the RD- or RD+ column of the DX.Y rows. The appropriate 8-bit data is communicated and the code_err is set to zero for any 10-bit data from either the RD- or RD+ column of the KX.Y rows. If any input of the form KX.Y, where K, X, and Y are indeterminate values, is provided, the code_err variable will have a value of one and the result will be undefined.

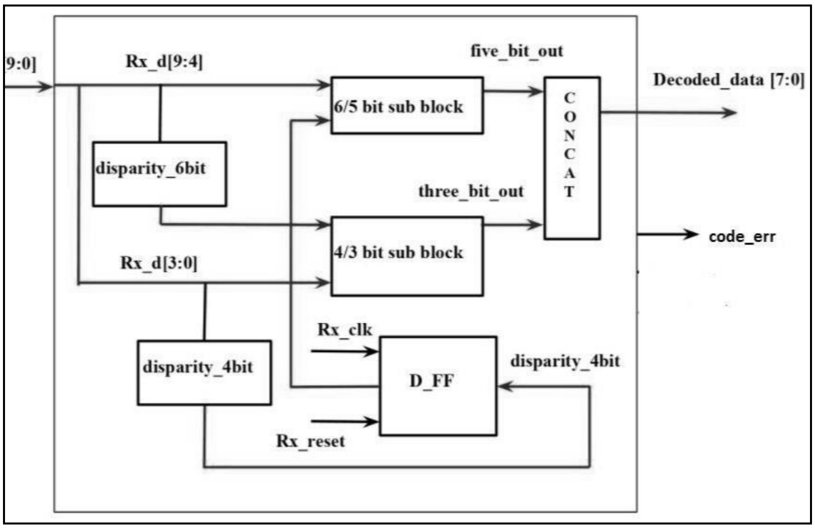


Figure 2 PCS Receiver Block Diagram

10b to 8b Decoding: The process involves converting a 10-bit input into an 8-bit output based on the specified valid data code groups and valid special code groups.

20b to 16b Decoding: The process of decoding involves converting a 20-bit input into a 16-bit output by using two sub-blocks that transform 10 bits to 8 bits each.

40b to 32b Decoding: The process involves converting a 40-bit input into a 32-bit output by using four sub blocks, each of which converts a 10-bit input into an 8-bit output.

3. Results And Discussions

We have executed the code using VCS (Verilog Compiler Simulator). The simulation results are shown in Figure 3, 4, 5, 6, and 7. The 8b/10b Encoder turns the 5-bit and 3-bit inputs into 6-bit and 4-bit outputs, respectively, on the first positive clock edge. This conversion is based on the information provided in Table 1 and 2. The combination of a 4-bit and a 6-bit value results in a 10-bit value at the next rising edge of the clock signal. The 16b/20b Encoder consists of two 8b/10b Encoding blocks. During the positive clock edge, two 8b/10b bit blocks operate concurrently, and bit conversion occurs according to Table 1 and 2. By combining the 4 and 6 bits from two blocks, two 10-bit outputs are generated during the following positive clock cycle. When we combine two 10-bit values, the resulting output is obtained at the next positive clock edge. The 32b/40b Encoder consists of four 8b/10b Encoding blocks. During the first positive clock edge, four 8b/10b bit blocks operate concurrently, and bit conversion occurs according to Table 1 and 2. By combining the 4 and 6 bits of two blocks, four 10-bit outputs are generated during the following positive clock cycle. The combination of four 10-bit values results in the final output at the subsequent positive clock transition.

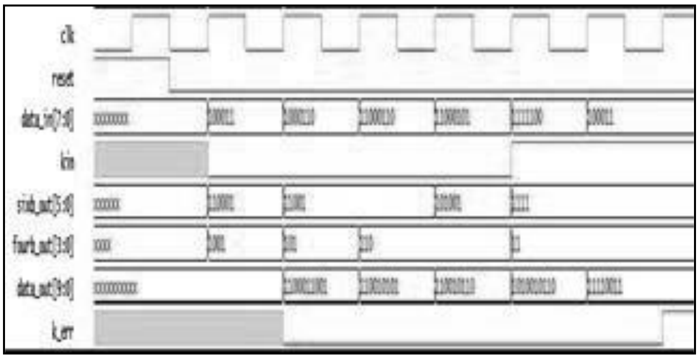


Figure 3 Encoder in Mode 00

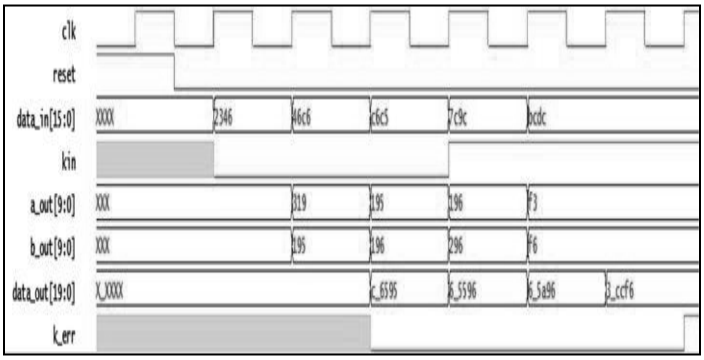


Figure 4 Encoder in Mode 01

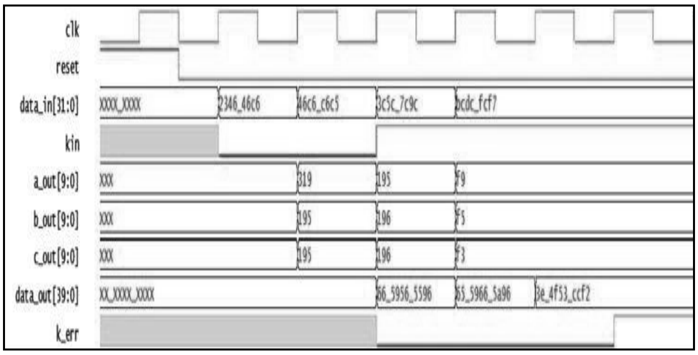


Figure 5 Encoder in Mode 10

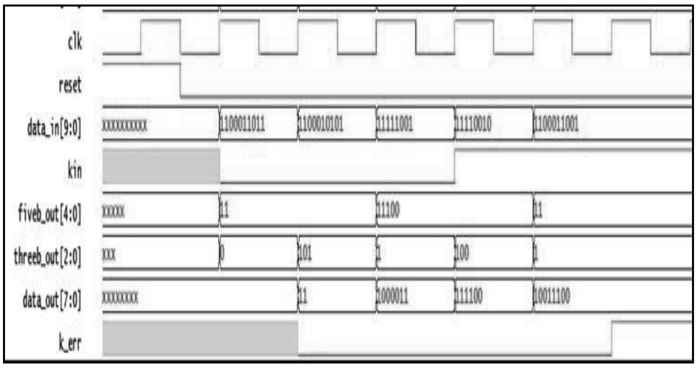


Figure 6 Decoder in Mode 00

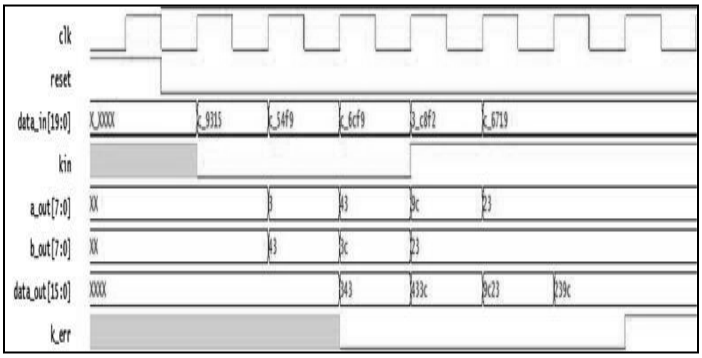


Figure 7 Decoder in Mode 01

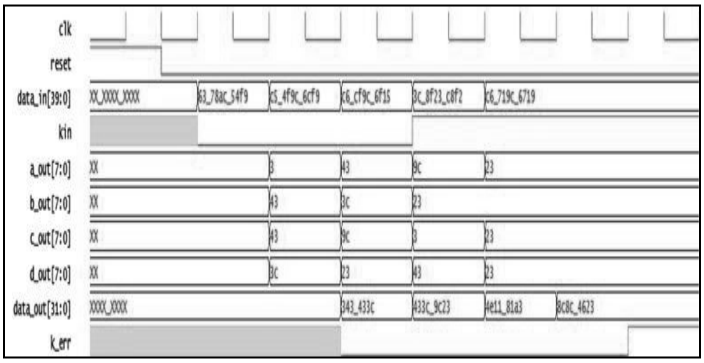


Figure 8 Decoder in Mode 10

The 10b/8b Decoder turns the 6-bit and 4-bit inputs into 5-bit and 3-bit outputs, respectively, on the first positive clock edge. This conversion is based on the specifications provided in Table 1 and 2. The combination of a 3-bit and a 5-bit sequence results in a 10-bit sequence at the subsequent positive clock edge. The 20b/16b Decoder consists of two 10b/8b Decoding blocks. During the positive clock edge, two 10-bit/8-bit bit blocks operate concurrently, and bit conversion occurs according to Table 1 and 2. The combination of the 3rd and 5th bits from two blocks results in two 8-bit outputs on the following positive clock cycle. The combination of two 8-bit values results in the final output at the subsequent positive clock edge. The 40b/32b Encoder consists of four 10b/8b decoding blocks. During the first positive clock edge, four 10b/8b bit blocks operate concurrently, and bit conversion occurs according to Table 1 and 2. The combination of the 3 and 5 bits from two blocks results in four 8-bit outputs at the following positive clock cycle. The combination of four 8-bit values produces the final result at the subsequent positive clock edge.

4. Conclusion

This study paper implemented the design of encoding and decoding mechanisms for 8b to 10b, 16b to 20b, and 32b to 40b based on disparity and valid data code groups, using CPRI and IEEE 803.2 standard as references. It may be used in digital communication to achieve DC balanced data transmission. It may be used in any context where PCS is necessary, as well as in situations that include high-speed data transmissions, Ethernet packet transfers, and fibre communication systems.

5. References

[1]. J. S. Turner, Terabit burst switching, Journal of High Speed Netw., 1999, 8(1), 3–16.

[2]. P. Gambini et al., Transparent optical packet switching: network architecture and demonstrators in the KEOPS project, IEEE Journal on Selected Areas in Communications, 1998, 16(7), 1245–1259

[3]. X. Widmer and P. A. Franaszek, A DC-balanced, partitioned block, 8B/10B Transmission code, IBM Journal of Research and Development, 1983, 27(5), 440–451.

[4]. S. Venkata Kishore, M.Srilatha, Mr.Y.S.Chakrapani, Design and implementation of high speed data transmission over dual independent aurora channels on one gtx dual tile usingvirtex-5 FPGA, International Journal of Scientific & Engineering Research, 2013, 4(12), 2229-5518.

[5]. M. Maadi, An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned Block, 8b/10b Transmission Code, International Journal of Electronics and Electrical Engineering, 2015, 3(2), 15-20.

[6]. P. Nallani, T. Vasudeva Reddy, FPGA implementation of low power serial to high speed data networks, International Journal on Recent and Innovation Trends in Computing and Communication, 2014, 2(8), 2377-2383.

[7]. V.R. Gad, R.S. Gad and G.M. Naik, Implementation of gigabit Ethernet standard using FPGA, International Journal of Mobile Network Communications & Telematics, 2012, 2(4), 62-70.

[8]. Kim M, Lee W, Yoon J, Jo O. Toward the realization of encoder and decoder using deep neural networks. IEEE Communications Magazine. 2019 May 13;57(5):57-63.

[9]. Lu Y, Li Z, He D, Sun Z, Dong B, Qin T, Wang L, Liu TY. Understanding and improving transformer from a multi-particle dynamic system point of view. arXiv preprint arXiv:1906.02762. 2019 Jun 6.

[10]. Kadel R, Paudel K, Guruge DB, Halder SJ. Opportunities and challenges for error control schemes for wireless sensor networks: A review. Electronics. 2020 Mar 19;9(3):504.