



Design And Validation Of Ethernet Physical Layer

¹Shivanand, ² Dr. Jeeru Dinesh Reddy,

¹Student, ²Assistant professor,

¹ Department of Electronics and
Communication,

¹ BMS College of Engineering Bengaluru, India

Abstract: The design and validation of an Ethernet physical layer protocol involving the implementation of Ethernet transmitter and receiver modules with CRC operation for data integrity verification is presented in this paper. The transmitter module is responsible for performing CRC operation on the data before transmitting it to the receiver side. On the receiver side, a similar CRC operation is conducted, and the received data is compared with the transmitted data to ensure data integrity and detect any potential data loss. This process aims to validate the successful transmission of data across the Ethernet network by confirming that the received data matches the transmitted data after undergoing CRC checks. The proposed protocol enhances data reliability and integrity within Ethernet communication systems.

Keywords— *Ethernet protocol, CRC, RTL Design, Xilinx Vivado, Verilog.*

I. INTRODUCTION

This work aims to design and validate an Ethernet physical layer using Verilog HDL for efficient communication between devices in a local area network (LAN). The focus lies on implementing transmitter and receiver modules responsible for framing data according to Ethernet standards and performing essential error checking mechanisms such as Cyclic Redundancy Checks (CRC). By adhering to these standards, we ensure interoperability with existing Ethernet infrastructure while maintaining reliable data transfer rates. The Ethernet frame structure consists of several components: preamble, Start-of-Frame Delimiter (SFD), destination address (dest_addr), source address (src_addr), length, payload, and Cyclic Redundancy Check (CRC). The proposed design consists of two primary components: a transmitter module responsible for preparing frames for transmission and a receiver module responsible for receiving frames from other devices on the network. Both modules will be implemented using Verilog HDL for simulation purposes followed by synthesis for FPGA implementation or ASIC design if required. This approach allows us to verify functionality at various levels of abstraction while maintaining flexibility in future modifications or enhancements. In the transmitter module, data is taken as input and formed into an Ethernet frame. If the payload size is less than the minimum length of 1500 bytes, the data enters a padding state. In this state, data is padded to meet the minimum length requirement. Once the frame is formatted, it undergoes CRC calculation. The CRC value is concatenated to the frame before transmission. Data is then sent bit by bit. On the receiver side, incoming bits are taken one at a time and converted from serial to parallel form. The received frame is reassembled and undergoes CRC calculation. If the calculated CRC value matches that transmitted with the frame, the data is considered valid and stored in a First-In-First-Out (FIFO) buffer for later retrieval. This process ensures reliable data transfer between devices connected via an Ethernet network.

II. PROPOSED METHODOLOGY

The Ethernet Top Module integrates an Ethernet Encoder and Ethernet Frame Extractor, facilitating efficient Ethernet communication. The Encoder constructs frames by calculating payload length, buffering data, and appending headers and CRC for error detection, managed by a transmitter state machine. The Frame Extractor validates and extracts received frames using CRC verification, header parsing, and payload storage, controlled by a receiver state machine. Internal buffers ensure smooth data flow and clock domain synchronization. Key features include high throughput, error detection, scalability, and protocol compliance. This modular design supports diverse Ethernet applications, ensuring reliable data transfer and easy integration with network systems.

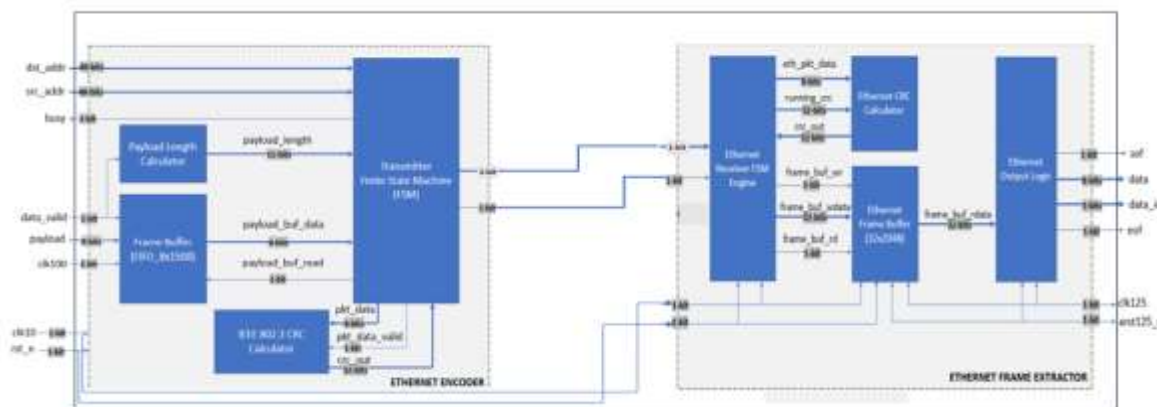


Fig 1. Block Diagram of Ethernet_Top Module

III. TRANSMITTER AND RECEIVER

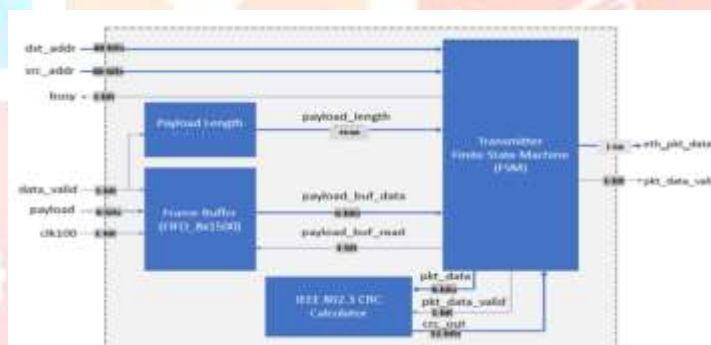


Fig 2. Block Diagram of Ethernet Transmitter Module

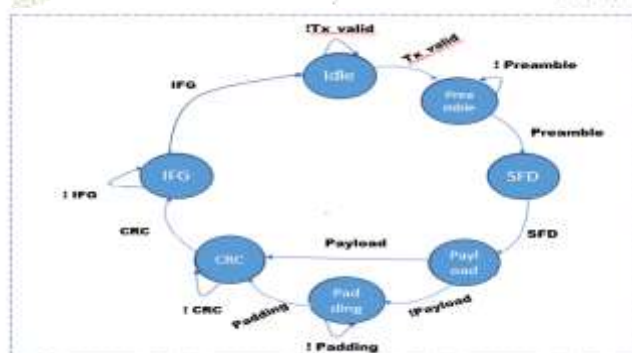


Fig2. Ethernet Transmitter Block Finite State Machine.

This section describes the methodology adopted for the design and implementation of the Ethernet transmitter module, as depicted in Figure 1. The design leverages a finite state machine (FSM) approach to handle the various stages of Ethernet frame assembly and transmission, ensuring efficient and reliable data communication.

Transmitter Module Architecture

The transmitter module is composed of several key components: Payload Length, Frame Buffer, IEEE 802.3 CRC Calculator, and the Transmitter Finite State Machine (FSM). Each of these components plays a vital role in the framing, padding, CRC calculation, and transmission processes.

3.1 Frame Buffer (FIFO)

The Frame Buffer is implemented as a FIFO (First-In, First-Out) memory structure with a capacity of 1500 bytes, aligning with the maximum payload size defined by Ethernet standards. This buffer temporarily stores the incoming data stream and facilitates sequential access to the payload data during frame assembly. The Frame Buffer outputs buffered data and read signals to the Transmitter FSM and the CRC Calculator

3.2 Ethernet frame

The transmitter module forms the ethernet frame by concatenating the destination address, source address, length, payload, and executed CRC values and then transmits the frame bit by bit towards receiver side over physical medium. It takes the incoming data stream and determines the total payload length, which is critical for frame assembly and of the payload length is less than required length it will do padding so that it should meet the required length.

3.3 IEEE 802.3 CRC Calculator

The CRC Calculator module performs the CRC operation as specified by the IEEE 802.3 standard that CRC-32. It computes the CRC value for the assembled frame, ensuring data integrity. The computed CRC value is then appended to the Ethernet frame before transmission. The CRC Calculator receives payload data from the Frame Buffer and outputs the CRC result to the Transmitter FSM.

Transmitter finite state machine

The Transmitter FSM is the core control unit that orchestrates the entire frame assembly and transmission process. It operates in several states, each corresponding to a specific stage in the frame construction. The FSM manages the sequence of operations required to construct and transmit Ethernet frames efficiently and reliably.

FSM remains in the Idle state until TX signal is asserted and the busy signal is 0. It moves to the PREAMBLE state where it consist of 56 bits of alternative 1 and 0 used for synchronization and next SFD this is the last byte of preamble indicates to the receiver that the actual data is on its way. The PAYLOAD or DATA here it consists of destination address, source address, length, actual data, in this state it calculates the total length of data if its less than the minimum length then it will go padding state where it will do padding to meet the minimum length of 1500 bytes and then it will go to the CRC state where it will do the CRC operation and the transmit the data. IFG state in this pause is essential to allow devices to prepare for the reception of the next frame. The IFG serves multiple purposes, including enabling receiver clock recovery, allowing devices to power up from low-power states, and providing a recovery time for the receiving device to be ready for the next frame.

Receiver Module Architecture

The Ethernet receiver module is a crucial component in the communication system, responsible for processing incoming Ethernet frames. It plays a vital role in ensuring data integrity and proper handling of the frame structure. The module consists of interconnected components that work together seamlessly to receive, verify, buffer, and output Ethernet frames. At the receiver side, the Receiver module is tasked with receiving data transmitted from the transmitter module. The external PHY (Physical Layer) receives data serially from the physical layer (cable) in serial format. These serials are then sent to the receiver module, which assembles them into data bytes.

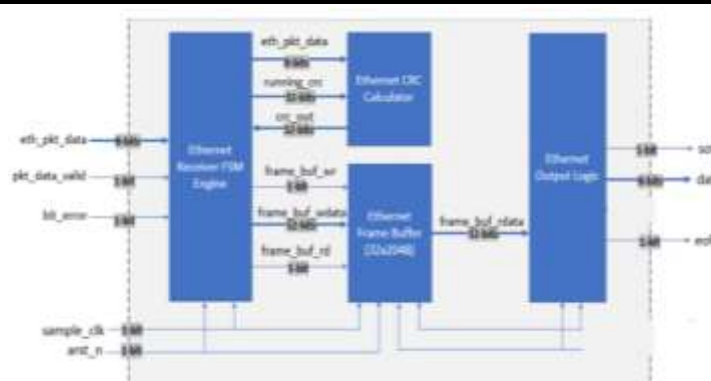


Fig 3. Block Diagram of Ethernet Receiver Module.

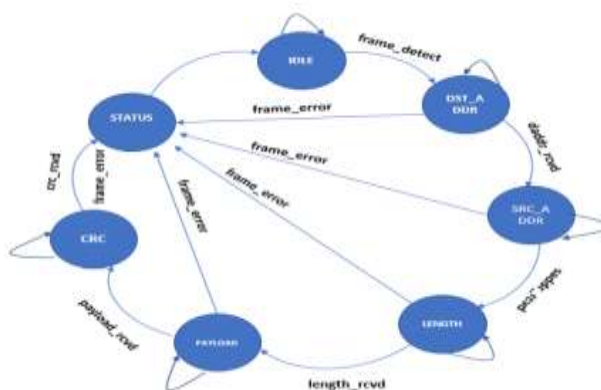


Fig 4. Ethernet Transmitter Block Finite State Machine.

Receiver finite state machine

The FSM ensures that frames are correctly received, validated, and processed according to the Ethernet protocol. Initially, the FSM resides in the IDLE state, where it awaits the detection of a new Ethernet frame. Upon frame detection (frame_detect), it transitions to the destination address state. This state handles the destination address and prepares for the reception of frame data. If any errors are detected during this phase (frame_error), the FSM moves to the STATUS state to handle the error. Otherwise, the FSM proceeds to the next state (source address) state upon successfully parsing the destination address. The FSM transitions to the LENGTH state, where it reads the length field to determine the size of the payload. Detection of any frame errors during this process would again shift the FSM to the STATUS state for error handling. Once in the LENGTH state, the FSM confirms the length field reception (length_rcvd) and moves to the PAYLOAD state to handle the reception of the frame payload data. After the entire payload is received (payload_rcvd), the FSM transitions to the CRC state, where it verifies the integrity of the frame by calculating and checking the Cyclic Redundancy Check (CRC). If any frame errors are detected during CRC verification (frame_error), the FSM moves to the STATUS state.

Through these transitions, the Ethernet Receiver FSM ensures that frames are accurately received and validated, maintaining the integrity and reliability of the Ethernet receiver module.

Ethernet Protocol and CSMA/CD

The Ethernet protocol uses a method called Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to control how data is sent over a shared network. In this system, devices first check if the network is clear (carrier sense) before they send any data. If two devices happen to transmit at the same time, a collision occurs. When this happens, both devices stop transmitting and send out a “busy signal” to indicate that the channel is occupied. Afterward, each device waits for a random amount of time before trying to send their data again. This process helps reduce the chances of collisions and makes sure that the network bandwidth is used efficiently, allowing for organized communication in local area networks (LANs).

IV. RESULTS AND DISCUSSION

The RTL Architecture for both the transmitter and receiver modules was designed and developed using Xilinx Vivado, cadence and ModelSim software. In this implementation, data transmission is handled in packet formats, adhering to the FIFO (First In, First Out) method during both transmission and reception processes. For the transmitter module, the process begins with the data to be transmitted being first stored in FIFO memory. This approach facilitates the generation of Ethernet packets, ensuring that data is organized and managed efficiently before being sent out. The minimum frame packet length is 64 bytes, which includes the data to be transmitted along with necessary Ethernet protocol headers. By utilizing FIFO, the transmitter module ensures that data is transmitted in the correct order and without any loss, adhering to the Ethernet protocol standards. Similarly, the receiver module employs a FIFO mechanism to handle incoming Ethernet frames. Upon receiving data, the receiver module processes it using a finite state machine (FSM) to manage the reception, validation, and buffering of the Ethernet frames. The FSM transitions through various states, such as IDLE, DESTINATION ADDRESS, SOURCE ADDRESS, LENGTH, PAYLOAD, CRC, and STATUS, to ensure that each frame is correctly received and verified. This systematic state transition ensures that the frames are accurately processed, with any detected errors handled efficiently.

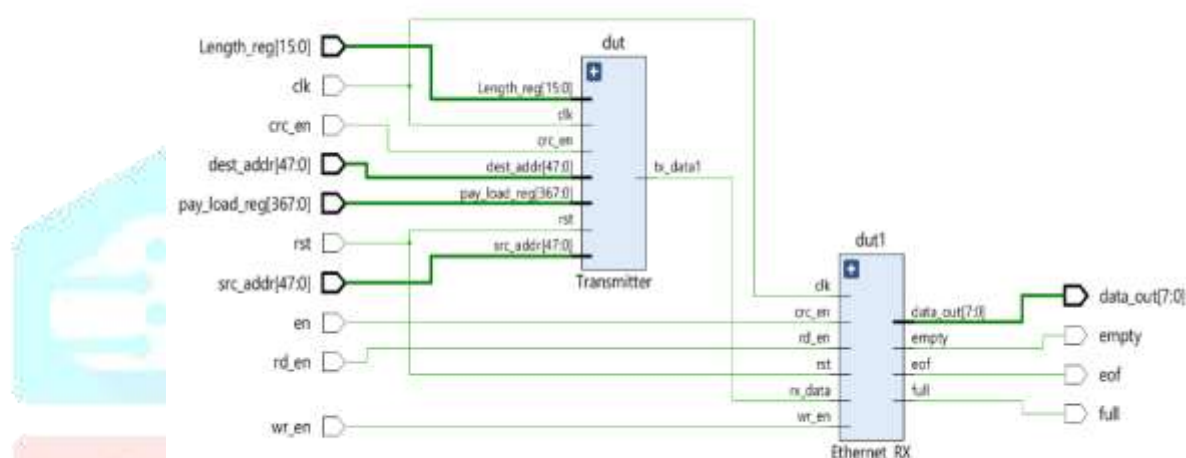


Fig 5. RTL Schematic of Ethernet_Top Block Diagram

4.1 Simulation Result

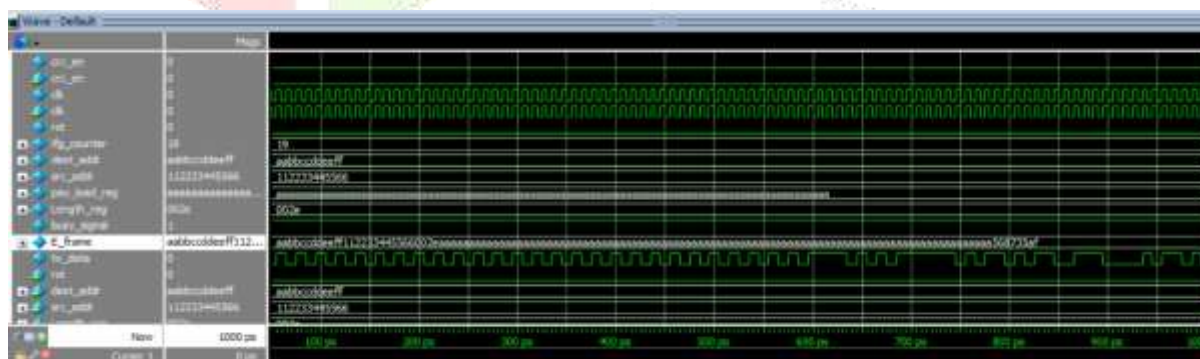


Fig 6. Simulation Result of Ethernet Transmitter

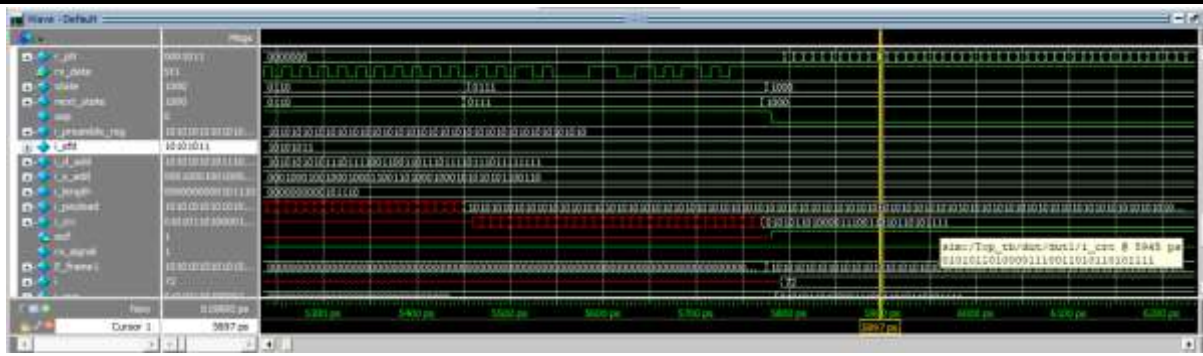


Fig 7. Simulation Result of Ethernet Receiver

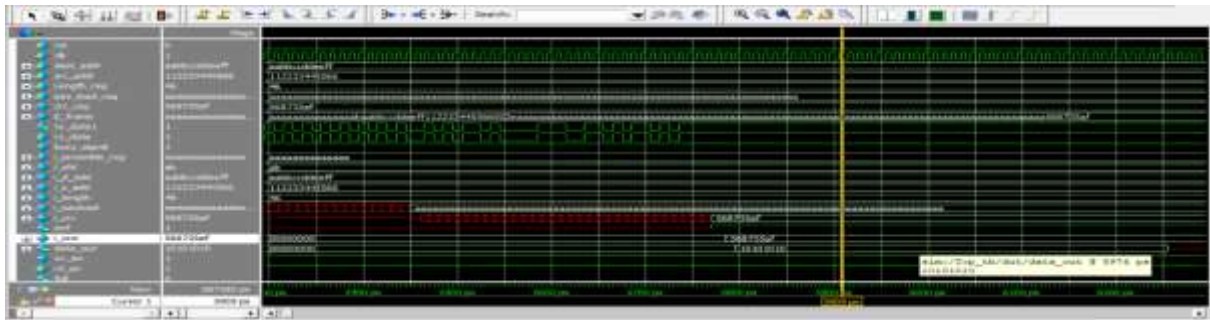


Fig 8. Simulation Result of Ethernet_Top

Area	7603 μm^2
Power	1.9 mw
Time	5.136 ns

Table 1. The synthesis report of the design

V. CONCLUSION

The Ethernet component is designed to process incoming payloads by calculating their length, buffering them, and appending CRC checksums in accordance with the IEEE 802.3 standard. This design is crucial for ensuring data integrity and is suitable for modern high-speed Ethernet applications. In parallel, the Ethernet Frame Extractor validates received Ethernet packets using a CRC calculator, storing verified data in its buffer, which ensures that only error-free data is processed.

The transmitter module successfully organizes data into Ethernet packets, maintaining a minimum frame packet length of 64 bytes. The receiver module, on the other hand, employs a finite state machine (FSM) to manage the reception, validation, and buffering of incoming Ethernet frames. The FSM's systematic state transitions ensure that each frame is processed accurately, with any errors detected and handled effectively. The design is implemented using the Cadence compiler targeting 45nm process technology. The design utilizes 7603 μm^2 area and the power consumption of 1.9mw and the setup time is 5.13ns.

VI. ACKNOWLEDGEMENT

We would like to acknowledge thank everyone and Assistant Professors of Dr. Jeeru Dinesh Reddy Dept. of ECE BMS college of engineering for his support in designing this project.

REFERENCES

- [1] A. R. Wadeyar, A. I. Rasheed and U. M. Roy, "Design and Simulation of Optical Fiber Communication Link by Ethernet Protocol using FPGA," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-5.
- [2] D. Bruschi, A. Di Pasquale, S. Ghilardi, A. Lanzi and E. Pagani, "A Formal Verification of ArpON – A Tool for Avoiding Man-in-the-Middle Attacks in Ethernet Networks," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 6, pp. 4082-4098, 1 Nov.-Dec. 2022.
- [3] K. Yadav, V. Mittal and V. Bhat, "Novel Implementation of a Configurable Gateway Between CAN and Ethernet Protocol For In-Vehicle Networks," 2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, 2020, pp.
- [4] D. Han, Y. Zhao and Y. Gong, "Design of Multi-Protocol Industrial Ethernet Security Monitor," 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China, 2021, pp.17-20.
- [5] Sung, G.M., Wang, H.K. and Li, Z.Y., 2018, May. Hardware design on FPGA for Ethernet/SONET bridge in smart sensor system. In 2018 7th International Symposium on Next Generation Electronics (ISNE) (pp. 1-4). IEEE.
- [6] Wang, H. K., Yu, C. P., Sung, G. M., & Li, M. W. (2018, October). Intelligent Packet Transformation and Transmission Between Ethernet and Optical Fiber Systems Based on a Field-Programmable Gate Array Board. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 4071-4076). IEEE [09] CALCULATION OF FIFO DEPTH - MADE EASY Compiled by Putta Satish.
- [7] K. L. V. Ramana Kumari, A. Surabhi, S. H. G and P. K. Reddy K, "Design and Verification of Ethernet Protocol," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6.
- [8] Y. Liu, "Design of Ethernet Interface for Liquid Level Collection in Conticaster Crystallizer," 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, 2023, pp.
- [9] C. Ouyang, Y. Qin and J. Zheng, "Research on Vehicle Network Analytic System Based on Ethernet Protocol Parsing," 2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2023, pp. 205-209.
- [10] Álvarez, L. Moutinho, P. Pedreiras, D. Bujosa, J. Proenza and L. Almeida, "Comparing Admission Control Architectures for Real-Time Ethernet," in IEEE Access, vol. 8, pp. 105521-105534, 2020.
- [11] X. Jiang, L. Chen and D. Fan, "Research on Safe Communication Architecture for Real Time Ethernet Distributed Control System," in IEEE Access, vol. 7, pp.
- [12] A. Pérez-Res, M. Garcia-Bosque, C. Sánchez-Azqueta and S. Celma, "Self-Synchronized Encryption for Physical Layer in Gigabit Ethernet Optical Links," in IEEE Access, vol. 8, pp. 89727- 89740, 2020.