

FPGA Implementation of Channel Codec for Optical Intersatellite Link Communication System

Irzal Zaini, Ida Bagus Krishna Yoga Utama, Yeong Min Jang

Department of Electronics Engineering

Kookmin University

Seoul, South Korea

irzal.zaini00@gmail.com; ibkyutama@ieee.org; yjang@kookmin.ac.kr

Abstract—The Optical Intersatellite Link (OISL) communication system is an important FSO technology with the increasing demand for high speed and high bandwidth internet. FSO channel condition is highly noisy therefore, a channel codec must be applied to the system to increase reliability. In this research, a FPGA board is utilized to implement channel codec for OISL systems. The main channel codec is based on a parallel arrangement of Reed-Solomon code for large data streams. CRC, Interleaver, Scrambler and LPC are added to improve the channel codec. The implementation results in low resource usage, but with high usage of memory blocks.

Index Terms—optical intersatellite link, FSO, channel codec, RS, FPGA.

I. INTRODUCTION

Optical Intersatellite Link (OISL) communication system is a Free-Space Optical (FSO) wireless technology that offers higher speed and higher bandwidth of data transmission compared to the traditional radio based system [1]. The current radio traffic is overly congested and limited, as a result, is insufficient medium for wireless communication. On top of that, the high amount of traffic makes radio based systems unreliable due to the signal interference, which is less affecting in FSO communication systems [2]. The FSO system is based on a laser as a transmitting medium hence, it also offers greater communication link distance and is very suitable to be used in space [3]. Most recent OISL system is used by Starlink by SpaceX, which provides wide coverage globally using space-to-space and space-to-ground link networks [4].

Unfortunately, the FSO communication link channel is highly noisy compared to the radio channel, caused by the Atmospheric Turbulence (AT). The laser used in the FSO system is easily distorted by the channel environment conditions in space and atmosphere such as dust particles, space debris and any other obstructing objects. The transmitting power of the laser will be significantly reduced at the receiver side because of the AT therefore, it is essential to include channel codec into the OISL communication system [5].

In this paper, it is presented the implementation of channel codec design that can improve the OISL data transmission reliability. The implementation is powered by Field-Programmable Gate Array (FPGA) development board, which is a highly flexible programmable device for designing such a channel codec system. The channel codec is implemented using a parallel arrangement of Reed-Solomon (RS) code, which

can handle large data streams concurrently. In addition to the RS as the main error correction code, Cyclic Redundancy Check (CRC), Interleaver, Scrambler, and Line Product Code (LPC) are added to secure the transmitted data further [6].

We designed a module interconnection system that connects each of the individual modules of the channel codec. These modules are: CRC-16, CRC-32, RS(32,16), RS(255,239), Interleaver, Scrambler, and LPC(24,16) [7]. The input and output bit sizes of these modules are different, therefore the need for module interconnection is necessary. Furthermore, the module interconnection supports easy modification to help future upgrade to the channel codec system. In addition, we designed system synchronization to ensure the proper timing of each module. The synchronization is utmost important in the channel codec system to avoid any bit overlap that could occur at the input and output nodes in the interconnection of the modules. The system synchronization is standardized and based on the global clock from the FPGA source clock.

II. METHODOLOGY

A. System Overview

The OISL channel codec implementation is conducted by creating a similar hardware topology using the FPGA development board. The FPGA used is Xilinx Kintex-7 XC7K325T, which supports data transmission speed up to 10Gbps thanks to its Small Form-factor Pluggable (SFP) port for fibre optic medium. The FPGA SFP ports are used as a substitute for the OISL laser module. The board also has two RJ45 Ethernet ports that are used to connect to the user terminals. Two PCs are used as user terminals, acting as input and output nodes. The user terminal runs Python code that runs UDP transmission.

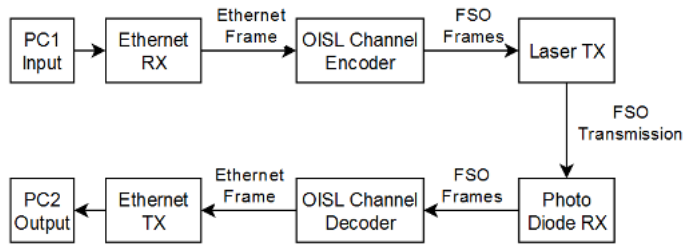


Fig. 1. OISL FPGA channel codec implementation block diagram.

The OISL FPGA channel codec implementation block diagram is shown in Figure 1. The hardware setup is shown in Figure 2. PC1 sends Ethernet Frame to the FPGA through the RJ45 Ethernet port. The FPGA then converts the Ethernet signal into serial bit data to be processed in the channel codec. The channel codec receives the bits and goes through a sequence of encoding processes, starting with Cyclic Redundancy Check (CRC) and then followed by parallel RS code with Interleaver. After that, the data is processed in the Scrambler and Line Product Code (LPC) unit and is ready to be sent to the fibre optic link. The fibre optic connection is made by using a loop from SFP port 1 to the SFP port 2. When the data goes back to SFP port 2 from the fibre optic link, it is processed in the channel codec decoder and then sent out from the FPGA to the PC2.

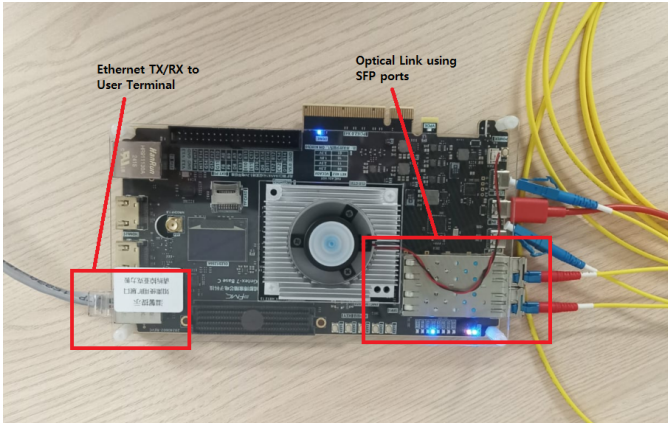


Fig. 2. The hardware setup of OISL implementation.

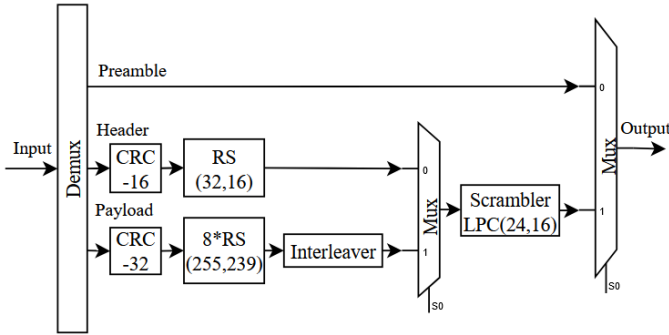


Fig. 3. Channel codec encoder architecture.

B. Channel Codec

The channel codec is a sequence of several block processes with configuration shown in Figure 3. The Ethernet Frame bits first are stripped into 3 parts which are: Preamble, Header and Payload. The Preamble is first checked for synchronization and then is put on the shift register. The Header is processed into CRC-16 and RS(32,16). The Payload is processed in a similar manner as the Header, however it is passed to CRC-32 and parallel arrangement of RS(255,239). For the

Payload, the encoded data from the RS is Interleaved. Before leaving the channel codec, additional processes are included: Scrambler and LPC(24,16). Here, the encoded Header and encoded Payload first concatenated and then passed through the Scrambler and LPC(24,16) process. Finally, the result of the channel codec is transmitted by the fibre optic module.

The individual modules of the channel codec: CRC, RS, Interleaver, Scrambler and LPC need precise synchronization to work together. Furthermore, the interconnection between modules also has to adjust according to the next input block bit size. Our contribution to this paper is designing a channel codec synchronization and interconnection functions to solve the issues mentioned above. The design is based on the standard of OISL, therefore compatible with the actual OISL system [6].

C. Cyclic Redundancy Check (CRC)

There are 2 CRC in the channel codec, the CRC-16 and CRC-32. The CRC-16 is used to add check bits to the Header, meanwhile the CRC-32 is used to add check bits to the Payload. The Header original Byte size is 14 Bytes, after the CRC-16 process the length will be 16 Bytes. The Payload original Byte size is 1908 Bytes, after the CRC-32 process the length will be 1912 Bytes. In the CRC process, the data are processed in bits serially. Therefore, for 1 cycle of Ethernet Frame, at least 15264 clock counts must be used to process it. The next process after CRC is the encoding at RS modules.

D. Reed-Solomon (RS) Code

The RS code is the main error correcting code that is used in the channel codec [8]. This allows the OISL communication system to locate and correct any errors occurring at the receiver side. The RS code adds parity symbols to the codeword before being sent to the transmission. The parity symbols are achieved through a polynomial based algorithm, which results in the ability to locate and correct errors to the original data. The RS decoder locates the error by extracting Syndromes out of received symbols. These Syndromes contain the information of error location and error magnitude in a polynomial equation. The next step is to reduce the value of Syndrome to find the location of the error symbol, and concurrently also the magnitude of the error symbol.

The Header and Payload are processed at different RS code channels. The Header is processed using shortened RS(255,239), which is RS(32,16), meanwhile the Payload is processed using parallel stacks of RS(255,239). The RS code for both Header and Payload will add 16 Bytes of parity symbols to the original data. The arrangement of parallel RS is to reduce processing time when handling a long stream of Payload data from input. The serial data bus from the CRC is Interleaved and also made parallel before entering RS modules. To operate the RS modules, a clock divider is necessary, since the CRC works in serial and the RS modules are in parallel.

E. Scrambler

The Scrambler unit is added to process the encoded Payload to scramble the bits in order to reduce burst error. At the receiver side, when a long sequence of zeros or ones are present, it could result in a bit of misreading. Therefore, to mitigate this, it is recommended to reduce the sequence of zeros or ones by scrambling it with a random predetermined constant, so the resulting bits will not be too monotone. The scrambler unit is processing data in 24-bit using a 25-bit scrambling sequence constant. This constant is always changing for each clock cycle, therefore the data are always randomly scrambled [7].

F. Line Product Code (LPC)

The LPC is put at the end of the channel codec to add additional parity checks to the final encoded data. The LPC used in the system is LPC(24,16), which means every 16-bit input will result in a 24-bit LPC codeword. The 16-bit input is arranged in a 4x4 bit matrix. The LPC parity bits are taken by XOR operation horizontally and vertically to the matrix. The resulting parity will be 8-bit size, hence the 24-bit LPC output [7].

G. Module Interconnections

The connection between modules are made by several connection techniques. The connections are made according to the input and output nodes of each channel codec module. The channel codec modules are: CRC-16, CRC-32, RS(32,16), 8 modules of RS(255,239), Interleaver, Scrambler and LPC(24,16). The connection from CRC-16 to RS(32,16) is serial to parallel 8-bit. Meanwhile, connection between CRC-32 to RS(255,239) is serial to parallel 64-bit. The Interleaver doesn't need any bit conversion. The Scrambler receives data from 8-bit RS(255,239) and after that, the LPC converts the data into 16-bit first to be processed at the final stage. The parallel output at LPC must be converted back into serial data to be sent to the fibre optic transmitter.

H. System Synchronization

To make the modules work together, synchronisation control must be implemented. We use a 2^{15} counter based on the global clock provided by the FPGA board. Each of the modules is assigned their own counter value. The counter value gives enable and reset time to the modules. The timing of these counters are based on the module data processing size. For the CRC-16 and CRC-32, 112 and 15264 counter values respectively. For the RS(32,16) and RS(255,239), 256 and 16384 counter values are needed. The Scrambler and LPC(24,16) need 24480 counter values.

III. TEST RESULTS

The test is conducted by observing the encoded data of its original messages. The OISL channel codec is compatible with standard Ethernet Frame or Jumbo Frame. In this section, we use 1931 Bytes test data consisting of Ethernet Preamble, Header and Payload.

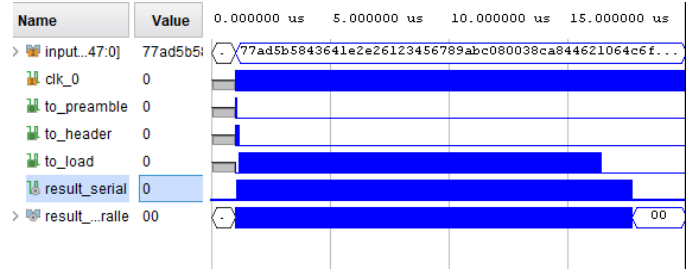


Fig. 4. Channel codec simulation result.

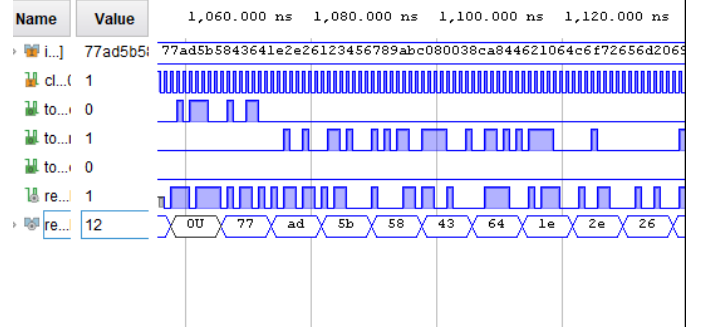


Fig. 5. Encoded data in magnified view.

Figure 4 shows the encoding result of the channel codec. The input data is first separated into 3 channels, Preamble, Header and Payload. The Payload is the longest data with 1908 Bytes size, while the Preamble and Header are only 9 Bytes and 14 Bytes respectively. The clock counter we used is 2^{15} , which counts up to 32768 clock counts. The system will follow the clock source of the FPGA.

Figure 5 shows the magnified view of the encoding process and the resulting parity symbols are shown in Figure 6. The resulting parity symbols combined with the original messages into 3117 Bytes of data, almost twice the size of the original message. This indicates a very low code rate, but can guarantee a very strong error correcting capability for the OISL system, especially for the adverse channel condition of FSO links.

The FPGA resource utilization shows quite low usage of logic hardware. However, the channel codec uses up to 30% of the Xilinx Kintex-7 XC7K325T memory. This is due to

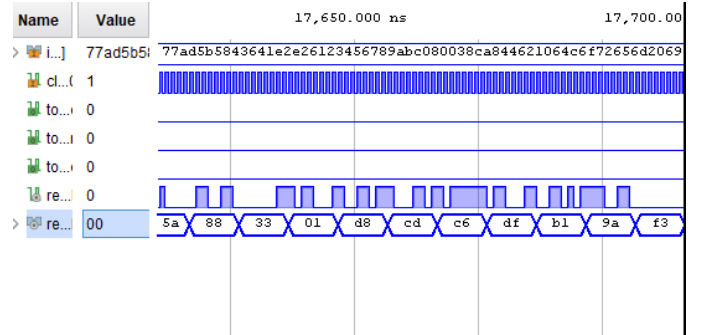


Fig. 6. The parity symbols at the end of original data.

TABLE I
KINTEX-7 XC7K325T RESOURCE UTILIZATION

Resource	Utilization	Available	Utilization %
LUT	4611	203800	2.26
LUTRAM	240	64000	0.38
FF	6658	407600	1.63
BRAM	134	445	30.11
IO	1	400	0.25

the implementation of multiple registers used in the channel codec, especially the RS modules. In the future, additional block memory such as Random Access Memory (RAM) would be a good idea to make this channel codec more optimal. The resource utilization can be seen in Table 1.

IV. CONCLUSION

An implementation of channel codec for Optical Intersatellite Link (OISL) is possible in FPGA based setup environment. The FPGA used has the appropriate specification to duplicate similar environment of OISL communication system. With FPGA, we can design the required channel codec and has the ability to easily modify the channel codec system thanks to FPGA re-programmability. Our channel codec design requires careful synchronization system to avoid data overlap or misreads. The design of module interconnection ensuring that each module of the channel codec can work properly together. The clock counter used is 2^{15} using the global clock source from the FPGA. The resource utilization showed very low logic

block usage, however a quite rather high memory usage due to the use of large count of registers in the channel codec. To mitigate this issue, it recommended to utilize Random Access Memory (RAM) that can be added externally to the FPGA system.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2022R1A2C1007884).

REFERENCES

- [1] M. K. Hasan, et al., "Optical camera communication in vehicular applications: a review," IEEE Transactions On Intelligent Transportation Systems, Vol. 23, No. 7, July 2022, doi: 10.1109/TITS.2021.3086409.
- [2] T. Nguyen, N. T. Le, Y. M. Jang, "Practical design of screen-to-camera based optical camera communication," 2015 International Conference on Information Networking (ICOIN), January 2015, doi: 10.1109/ICOIN.2015.7057916.
- [3] Z. Xia, et al., "Inter-satellite link channel characterization of laser communication systems," 2023 IEEE 11th International Conference on Information, Communication and Networks, 2023, doi: 10.1109/ICIN59530.2023.10393617.
- [4] T. Duan, V. Dinavahi, "Starlink space network-enhanced cyber-physical power system," IEEE Transactions on Smart Grid, Volume: 12, Issue: 4, July 2021, doi: 10.1109/tsg.2021.3068046.
- [5] A. Imran, et al., "Channel estimation of massive MIMO FSO communication system using deep attention residual U-Net," ICT Express, September 2024, doi: 10.1016/j.icte.2024.09.012.
- [6] M. Butterfield, "Space Development Agency optical intersatellite link (OISL) standard," January 2022.
- [7] "Optical high data rate (HDR) communication—1064nm," Research and development for space data system standards, December 2018.
- [8] C.K.P. Clarke, "Reed-Solomon error correction," R&D White Paper, July 2002.