Serial communication protocol for FPGA-based systems

V K Ivanov¹ and E V Nosov²

¹Peter the Great St. Petersburg Polytechnic University, Polytechnicheskaya 29, St. Petersburg, 195251, Russia
²Institute of Applied Astronomy of the Russian Academy of Sciences, Kutuzova Embankment 10, St. Petersburg, 191187, Russia

E-mail: viva1444@bk.ru

Abstract. The article presents results of development of communication protocol for UART-like FPGA-systems. Developed communication protocol supports asynchronous oversampled signal transmission, with 4b/6b encoding for DC-balance maintaining. Algorithm of CPLD-based receiver of signals forming by this protocol is described. This development is relevant for the task of continuous transmission of several control signals over a fiber-optic line, when the use of ready-made UART devices is unjustified. The presented protocol and algorithm are easily implemented on the FPGA and can be used in devices where it is important to minimize the number of elements used. Also, the protocol contains encoding that allows the transmission of low-frequency signals along a line with DC-coupling.

1. Introduction

In the infocommunication system it is often necessary to make asynchronous data transmission. The node of digital devices that provide such a transmission is called universal asynchronous receiver-transmitter (UART) [1]. To transfer data through such a node, special data transfer protocols (for example, Ethernet or RS-232) are created [2]. Transmission occurs by serializing data and creating a message according to a specific protocol, which is transmitted to the receiver.

![Figure 1. Typical message chart for UART protocols](image)

In addition to standard solutions, various private solutions are being developed, both for the implementation of standard protocols and for their modification. For example, various tasks on the implementation of UART on FPGAs are discussed in [3-5], the use of FPGAs for operation by the physical Ethernet layer is discussed in [6-8]. As we can see, there are many different developments in implementing UART systems on FPGAs so this is actual task nowadays.

In this work, the problem of transmitting several signals over an optical line with an SFP interface was solved. To do this, these signals must be appropriately encoded on the transmitter side and then restored to the receiver. The RS-232 protocol is not suitable for this task, since due to transmission over the fiber optic line it is necessary to provide the same number of zeros and ones for DC coupling.
For our task, you can use the Ethernet protocol and the corresponding turnkey solutions on the FPGA, however such a solution would be redundant, since the Ethernet protocol is designed to transfer a large amount of data to the CPU, which makes the protocol and the device itself more complicated, it requires the use of an FPGA with a large number of logic elements operating at high frequency. At the same time, to solve our problem, you can create a special data transfer protocol, which will reduce the requirements for the device. In particular, instead of standard FPGAs, you can use CPLDs (Complex Programmable Logic Device) that contain a small number of logical elements and have internal memory. Another difference is that Ethernet is designed to receive packets of data that arrive at different times, which makes it necessary to control the flow of data. In our case, the messages are transmitted continuously, which creates its own specifics of the operation of the device. For example, an under / overrun problem, when the processor does not have time to read the information or when the processor processes the information too quickly - there is no such problem in the FPGA device, so messages can be sent continuously.

Summary, the development uses some ideas of popular protocols to create a special protocol for asynchronous continuous information transfer. Also, serialization algorithm was developed so we can transfer several discrete signals with this protocol. For example, such a system can transmit several meanders of different frequencies on a single fiber optic link.

2. Materials and methods
The developed protocol was based on known UART protocols. Then programs were written in the Verilog language in the ISE Design Suite for the FPGA for the transmitter and the receiver, in which was laid the algorithm for creating the parcels and for their decryption. Checking the code was carried out in the simulator and then on the assembled device. The device was based on CPLD Xilinx XC2C256 FPGA, a 100 MHz oscillator and a SFP transceiver for transmitting information via fiber-optic links. Evaluated the stability of the algorithm, restrictions on the frequency of transmission. To track the stability of the work, an error tracking algorithm was provided.

![Figure 2. Experimental setup](image)
3. Results

As a result, a data transfer protocol was obtained, which provides asynchronous continuous data transfer and an FPGA algorithm was created for transmitting and receiving data using this protocol. The following is a message chart and waveform of signal formed by this protocol:

![Message chart for developed protocol](image1)

**Figure 3.** Message chart for developed protocol

![Waveform of signal formed with developed protocol](image2)

**Figure 4.** Waveform of signal formed with developed protocol

Unlike, for example, the RS-232 protocol, a sync word of a fixed length is added here instead of the idle state. The receiver performs synchronization by tracking the change in the signal level, that is, on the signal front. Each bit is transmitted several times - thus providing stability to the difference between the clock frequencies of the receiver and transmitter (simulation has shown that the difference can reach units of percent). Number of bits in sync word should be bigger than number of data bits so receiving system can always determine where data is and where is a sync word. This condition can be defined as

\[ B_{\text{sync}} > B_{\text{data}} \]  

(1)

Since the algorithm was developed for a low-frequency device, the DC coupling problem arises, so coding was used. Developed encoding based on 8b/10b encoding ideas, that shown in [9]. Since we’re using continuous data flow we don’t need algorithms for running disparity control. Table for “4b/6b” encoding is shown below. The idea of this encoding is to put any of 4-bit word to one-to-one
correspondence with 6-bit code with equal number of “zeros” and “ones”. Number of data bits in message defined as

$$B_{data} = \begin{cases} 
N_{signal} + 2, & \text{if } N_{signal} \text{ is even;} \\
N_{signal} + 3, & \text{if } N_{signal} \text{ is odd;}
\end{cases}$$

(2)

where $N_{signal}$ is number of bits before encoding. Expression in brackets in (2) should be even.

### Table 1. Table of 4b/6b encoding

| Input  | 6b code | Input  | 6b code |
|--------|---------|--------|---------|
| 0000   | 000111  | 1000   | 101001  |
| 0001   | 001011  | 1001   | 101010  |
| 0010   | 001101  | 1010   | 101100  |
| 0011   | 001110  | 1011   | 110100  |
| 0100   | 010110  | 1100   | 111000  |
| 0101   | 011010  | 1101   | 100101  |
| 0110   | 011100  | 1110   | 110010  |
| 0111   | 011001  | 1111   | 100110  |

At the source of the message and the receiver, the clock frequency may be slightly different. Oversampling is used to fix this problem. Each bit is transmitted several times, that is, the duration of each bit is increased a certain number of times. This reduces the transmission rate, but it simplifies the synchronization of the transmitter and receiver. In the developed algorithm, the receiver reads the input state from the middle of the bit and synchronizes on the signal front. During the simulation, it was found that the frequencies of the receiver and transmitter can differ by a few percent. Similar results are presented in [10]. According to this paper, we have formula for maximum period error:

$$\Delta_{max} = \frac{S/2 - 1}{SB_{data}}$$

(3)

For oversampling ratio $S = 10$ and $B_{data} = 6$ we have $\Delta_{max} \approx 6.67\%$. For example, ASDM oscillators, which we used in experiment, have 100 ppm frequency stability [11].

Overall length of message can be calculated by next formula:

$$T_m = T_b S \left( B_{sync} + B_{data} \right)$$

(4)

where $T_b$ is length of oscillator period, $S$ is number of samples (oversampling multiplier). From (4) we have expression for sampling frequency:

$$\Delta F = \frac{1}{T_m}$$

(5)

Receiver was implemented on CPLD. The Verilog consists of if-else and case statements. Sample counter for oversampling, shift register and output register are provided in the device. The input sequence is written to the shift register, and if its contents match one of the values in Table 1, the corresponding value is written to the output register. Information about the resources required by the implementation of the algorithm on CPLD XC2C256-7TQG144C is given in Table 2.
Table 2. Resources, needed for algorithm implementation.

| Resources         | Amount          |
|-------------------|-----------------|
| Macrocells        | 121/256 (48%)   |
| Pterms            | 401/896 (45%)   |
| Registers         | 101/256 (40%)   |
| Function Block Inputs | 276/640 (44%) |

4. Discussion

With the help of the obtained algorithm, it was possible to realize the continuous transmission of several digital signals along one line. Naturally, the transmitter clock frequency must be higher than the frequency of the transmitted signals. The exact maximum limit for the periods of transmitted signals is indicated in the formula:

$$T_{signal} < T_m$$

or

$$F_{signal} < \frac{1}{T_m}$$

Also, we can calculate the spectrum efficiency [12, 13]:

$$R = \frac{N_{signal}}{\Delta F}$$

where $R$ is a bit rate, $\Delta F$ is a bandwidth determined by sampling frequency. From (4) we have:

$$\frac{R}{\Delta F} = \frac{N_{signal}}{T_m} = N_{signal} \frac{1}{T_m}$$

which simply means that spectrum efficiency is equal to number of transmitted bits in one message. But we should remember that increasing the number of signal bits lead to increasing the length of message and decreasing the sampling frequency.

5. Conclusion

A data transfer protocol was obtained that provides asynchronous continuous transmission of several data streams over a fiber-optic line and an algorithm for an FPGA for transmitting and receiving data using this protocol was created. A table has been developed for 4b / 6b coding. Expressions for protocol parameters was shown. The resulting algorithm was successfully tested in the simulator and on the assembled device.

The protocol is intended for systems where it is necessary to transmit several signals over a single optical line. The resulting protocol is quite simple to implement on CPLD, which simplifies the creation of a device for working with this protocol.

References

[1] Bell C G, Mudge J C and McNamara J E 1978 Computer Engineering: A Dec View Of Hardware Systems (Massachusetts: Digital Press)
[2] Determining Clock Accuracy Requirements for UART Communications 2003 (Maxim Integrated Products)
[3] Amrutha G and Nayana D K 2018 FPGA Implementation of UART with Single Error Correction and Double Error Detection (UART-SEC-DED) Int. J. of Engineering &
Technology 7 3.12

[4] Patel N and Patel N 2013 VHDL implementation of UART with BIST capability 4th Int. Conf. on Computing, Communications and Networking Technologies (ICCCNT) (IEEE)

[5] More J et. al. 2015 FPGA Implementation of Universal Asynchronous Transmitter and Receiver Online International Conference on Green Engineering and Technologies (IC-GET) (IEEE)

[6] Salem E et. al. 2018 FPGA implementation of 1000base-x Ethernet physical layer core International Journal of Engineering & Technology: Science Publishing Corporation 7 4 2106-2112

[7] Sung G-M, Wang, H-K and Li Z-Y 2018 Hardware design on FPGA for Ethernet/SONET bridge in smart sensor system 7th IEEE Int. Symp. on Next-Generation Electronics (ISNE 2018)

[8] Alachiotis N, Berger S A and Stamatakis A 2010 Efficient PC-FPGA Communication over Gigabit Ethernet. 2010 10th IEEE Int. Conf. on Computer and Information Technology

[9] Schouhamer Immink K A 2004 Codes for Mass Data Storage Systems (Eindhoven: Shannon Foundation Publishers)

[10] Raj M 2017 UART Receiver Synchronization: Investigating the Maximum Tolerable Clock Frequency Deviation Indian Journal of Science and Technology 10 25

[11] Mems Clock Oscillator 2018 (Abracon LLC)

[12] Ovsyannikova A S, Zavjalov S V and Volvenko S V 2018 On the Joint Use of Turbo codes and Optimal Signals with Increased Symbol Rate 10th Int. Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)

[13] Zavjalov S V, Volvenko S V and Makarov S B 2016 A Method for Increasing the Spectral and Energy Efficiency SEFDM Signals IEEE Communications Letters 20 (12) 2382 – 2385