Low-Latency Ultra-Wideband High-Speed Transmission Protocol Based on FPGA

Jinxian Hu, Jinfeng Wang and Rengang Li
The 723 Institute of CSIC, Yangzhou 225001, China
Email: lichunlai117@163.com

Abstract. Aiming at the shortcomings of current high-speed serial transmission protocol including high latency and low effective transmission bandwidth, this paper designs a low-latency, ultra-wideband high-speed serial transmission protocol based on FPGA. Based on homologous system clock, data scrambling and synchronous frame synthesis are used to complete the data integration, so as to increase the effective bandwidth of data transmission. The receiving end reduces the data transmission latency by using the internal logic of FPGA and the buffer pool built by the register group together with status machine. The protocol is simple and universal, which can be applied to the FPGA platform of any OEM. Experiments show that the protocol can effectively reduce data transmission latency, improve the effective bandwidth of data transmission, and monitor the health and error code status of each data transmission channel in real time.

Keywords. FPGA; high-speed transmission protocol; low-latency; ultra-wideband.

1. Background
In recent years, with the rapid development of communication, radar and electronic countermeasure techniques, demand for data exchange with low-latency and high bandwidth increases gradually. The traditional parallel transmission methods meet great difficulties in pipeline cost, interface complexity, mutual interference and transmission distance etc. [1]. High-speed serial transmission/receiving technique based on FPGA provides a proper solution [2]. It is also commonly used in experiments for the distribution of trigger and control systems and for data transfer [3]. But current protocols based on transceiver inside FPGA such as Ethernet, PCIE, RapidIO and JESD204B focus on different features and the latency is relatively high. Besides, their transmission bandwidth is limited by protocol coding and channel binding quantity, so it is difficult to find a low magnitude transmission protocol which is able to achieve both low latency and high effective bandwidth. This paper designs one kind of universal transmission protocol with low-latency and ultra-wideband based on FPGA high-speed serial transceiver. Taking XilinxFPGA as implementation platform, the protocol implementation principle and critical techniques are introduced in detail. At last, the proposed protocol is validated to be able to effectively reduce data transmission latency and improve the effective bandwidth of data transmission by experiments.

2. Introduction and Feature of Protocol
The proposed protocol mainly consists of 5 modules including high-speed serial transceiver module, synchronous frame synthesis module, channel synchronization module, transmitter reset control module and receiver reset control module. High-speed serial transceiver module mainly conducts transmission and receiver of original data and provide benchmark clock for system. Synchronous
frame synthesis module. Synchronous frame synthesis module is employed by data transmitter to make scrambling to transmitted data and synthesize synchronous frame for data synchronization. Channel synchronization module is employed by data receiver, so as to analyze the synchronous frame and conduct data descrambling and synchronization in each data channel. Transmitter reset control module and receiver reset control module are mainly for reset and initialization of transceiver and system logic, so as to guarantee that the system power-on is in confirmed status. The logic frame of proposed solution can be found in figure 1.

![Logic frame](image)

**Figure 1.** Logic frame.

The features and advantages of proposed protocol are as follows.

1. **High effective transmission bandwidth:** channel quantity \(N \geq 2\), effective transmission bandwidth increases with increase of channel quantity. Line rate is decided by up limit of FPGA transceiver rate.

2. **PRBS data scrambling:** Guarantee of DC balance, transmission bandwidth is not occupied [4].

3. **Low latency:** Latency of transmission/receiving is no more than 125ns.

4. **Low magnitude protocol:** Resources consumption inside FPGA is low.

5. **Real-time channel status monitoring:** Health and error code status of each channel can be monitored in real-time.

### 3. Critical Techniques

The reference clock for transceiver operation is provided by uniform clock source so as to guarantee there is no accumulation without clock compensation system. The \(N-1\) pairs of transceivers of transceiver group conduct data transmission after PRBS scrambling and PRBS code type is selectable. The last pair of transceivers conducts transmission of synchronous frame and control information. As mentioned above, the bandwidth utilization rate of data transmission \(B\) is as follows.

\[
B = \frac{(N-1)}{N} \quad N \geq 2
\]

So increase of \(N\) leads to higher effective transmission bandwidth of protocol. Data transmission mode of pseudo random sequence scrambling may not occupy extra transmission bandwidth. The effective bandwidth utilization rate of the protocols employing 8B/10B coding such as JESD204B and RapidIO etc is fixed as 80%. All the internal control of each critical module of protocol employs status machine so as to separate control flow and data stream. In the data stream part, stream progression is
strictly controlled and shift buffer pool built by register group and search table is employed for data buffer alignment instead of FIFO, which guarantees the low latency feature of protocol. Each critical technique will be introduced in detail as follows.

3.1. Synthesis of Synchronous Frame

Shown as figure 2, the first N-1 routes of original data are processed by PRBS scrambling and sent to transceivers relevant to each channel in synchronous frame synthesis module. Frame synchronization module adds synchronous frame head and information to the scrambled data and generates synchronous data frame as per data format in figure 2 and sends it to frame synchronous transceiver. The synchronous data frame consists of scrambled data and frame head information. The frame head consists of 3 clock rhythms including synchronous frame head 0, synchronous head 1 and synchronous frame information. The synchronous frame head 0 and synchronous head 1 are to mark the start of synchronous frame and the specific content can be defined flexibly. Synchronous frame information is to extend protocol function and send necessary control command such as scrambling synchronous control command, which is for initialization PRBS descrambling code type in receiver. The succeeding data in synchronous data frame is the copy of data in current clock rhythm in each data channel for continuous 4 clock rhythms as figure 2, which is for synchronization of each channel in receiver.

![Figure 2. Block diagram of data transmission & receiver module and synchronous frame format.](image)

3.2. Channel Synchronization

Channel synchronization module mainly consists of 2 parts including synchronous analysis module and data synchronization module as figure 3. Transceiver group in receiver recovers the serial data in transmission channel to N-1 routes of scrambled data and 1 route of synchronous frame. The synchronous frame is sent to synchronous frame analysis module. The status machine in that module will obtain data synchronous information and scrambling mark signal. Data synchronization module utilizes the data synchronous information to conduct data alignment to received scrambled data. The
data after alignment is sent to data descrambling module for data recovery. The scrambling mark signal is for initialization of descrambling code word.

![Block diagram of receiving data module](image)

**Figure 3.** Block diagram of receiving data module.

The synchronous flow in synchronization analysis module is conducted by status machine as figure 4. After reset of status machine, it will search for synchronous frame head in data stream of synchronous frame. If there is time-out, it will move the shift buffer pool inside module to conduct 1bit slide of search data window, and the 8bit counter “shift_value” inside shift buffer pool will be added by 1 and the mode of searching for synchronous frame head 0 will be entered again. If the head 0 is found, status machine enters the mode to wait for synchronous frame head lock-up and keeps tracking frame head position. When quantity of continuously found frame heads reaches preset threshold, it enters synchronous frame lock-up mode. In synchronous frame lock-up mode, status machine keeps monitoring the data status of frame synchronization channel and judges frame synchronous head 0 and 1 are correct or not at each fixed position. If both are correct, it provides synchronous frame analysis success mark and start mark, and sends them to data synchronization module to start data synchronization. Otherwise relevant error counting status will be entered. When continuous error quantity is beyond preset threshold, the synchronization flow will be restarted.

The synchronous flow in data synchronization module is conducted by status machine and each channel conducts data alignment corresponding to synchronous frame independently as figure 5. After frame synchronization accomplishment, each status machine waits for the arrival of initial position of next synchronous frame. After entering data comparison flow, it compares the data in continuous 4 rhythms in certain data channel with data of corresponding position in synchronous frame. If the comparison result is the same, it means alignment for data in current channel with synchronous frame is accomplished. If there is data mismatch, mismatch counter will be added by 1. If the mismatched counting value is beyond preset threshold, the buffer pool will slide to change 1bit latency of data search window and counter “shift_value” will be added by 1 and above loop is entered again. When any channel accomplishes data alignment, status mark signal will be provided and status machine...
enters data monitoring period. According to the comparison to data of each link (it should be the same when there is not error code), it monitors the health and error code status of each channel.

![Diagram of synchronous frame analysis flow]

Figure 4. Analysis flow of synchronous frame.

### 3.3. Shift Buffer Pool

As core module to achieve frame synchronization and data synchronization in receiver, the shift buffer pool conducts final multi-channel data alignment through change of data latency inside buffer pool. The module is usually accomplished by FIFO and other memory components inside FPGA with high latency. In order to guarantee the low latency feature of proposed protocol, selector and register group whose total length is 295bit (255bit + 40bit) are employed. The 40bit data output by transceiver achieves data latency of 0–255UI (1UI=1s/(transceiver line rate (G)*10^9)) by buffer pool. 40bit data achieves the shift from left to right at the rate of 40bit each clock rhythm. The latency value is determined by above mentioned counting value “shift_value” and 40bit data is selected from shift chain for output.

Buffer pool employs 8-level 1-in-2 MUX (selector) to achieve data selection. Each bit of shift_value is corresponding to selector of each level. The top bit of shift_value (the seventh bit) is corresponding to selector LEVEL8 and decreases the selection room of whole latency chain from 295bit (2^8-1+40) to 167bit (2^7-1+40). Similarly, the sixth bit of shift_value is corresponding to selector LEVEL7 and decreases the selection room of latency chain from 167bit (2^7-1+40) to 103bit (2^6-1+40). Consequently, according to the same principle, the last selector LEVEL1 will output 40bit data after
latency. The whole flow see figure 6. 2-stage register is employed in selection chain to guarantee the time sequence of logic link. The finally obtained data latency in that module can be expressed as follows.

\[
\text{BUFFER_DELAY} = 2 \times \text{CLK\_PERIOD} + \text{shift\_value} \times 1\text{UI}
\]  

(2)

where \(\text{shift\_value}\) is influenced by link latency and interface bit width setting of transceiver.

**Figure 5.** Data synchronization flow.

**Figure 6.** Output selection logic of shift buffer pool.
4. Performance Validation
The features of proposed protocol will be tested by experiment. Xilinx FPGA xcv690t-fft1927 is employed as estimation component [5]. The protocol is accomplished by 24 groups of transceivers. 23 groups are for high-speed data transmission and 1 group is for synchronous frame transmission. The effective data rate is 23/24*100%=95.83%. Transceiver line rate is 10.24Gbps and total transmission bandwidth can be up to 235.52Gbps. The transceiver interface data bit width is 40 and internal logic operation clock is 256 MHz. FPGA resources occupation rate see following table 1. We can see that the proposed solution has low LUT resources occupation. As public resources, BUFG and MMCM clock can be shared by other modules.

| Resources | Occupied | Available | Occupation rate |
|-----------|----------|-----------|-----------------|
| LUT       | 15010    | 433200    | 3.4%            |
| LUTRAM    | 54       | 174200    | 0.03%           |
| RIGISTER  | 21368    | 866400    | 2.46%           |
| GTH       | 24       | 80        | 30.00%          |
| BUFG      | 5        | 32        | 15.63%          |
| MMCM      | 2        | 20        | 10.00%          |

Then latency performance of proposed protocol is tested. Transmitter employs one 40bit counter as data source to send data to each route at the same time. Transceiver mode is set as PMA self-loopback mode [6]. The receiver can receive the delivered counting value in synchronization and transmission pipeline latency can be omitted. Transmission latency can be obtained by calculation of the counting difference between transmitter and receiver. The experiment snapshot is as follows.

![Figure 7](image1)

**Figure 7.** Test to latency performance of protocol.

From the above figure 7, we can obtain the transmission latency of proposed protocol is: DELAY_VALUE(30)*CLK_PERIOD (3.906 ns)=117.8125 ns. Due to the latency uncertainty caused by transceiver clock frequency division ambiguity, each power-on latency may lead to certain difference [7]. After long-term test, it is obtained that the time jitter caused by ambiguity may not exceed 4 ns, so the maximum latency of proposed protocol will not exceed 125 ns.

The transmission characteristics of proposed protocol is validated by mutual transmission between backboards of 2 PCBs. The counter same as above mentioned is employed. The data received by 23 LANEs in receiver is shown in figure 8.

![Figure 8](image2)

After PRBS decoding, each LANE receives same counting value and the proposed protocol makes automatic compensation to latency difference between each LANE, which achieves effective data synchronization alignment and guarantees the correct and effective transmission of wideband data.
Figure 8. Test to data transmission performance of protocol.

5. Conclusion
Aiming at the defects of universal protocols including high latency and limited transmission bandwidth, this paper designs one kind of low-latency, ultra-wideband universal high-speed serial transmission protocol based on FPGA high-speed serial transceiver. Taking Xilinx FPGA as implementation platform, this paper makes detailed demonstration to relevant implementation principle and critical techniques. Data scrambling and synchronous frame synthesis are used to complete the data integration, so as to increase the effective bandwidth of data transmission. By using the internal logic of FPGA and the buffer pool built by the register group together with status machine, transmission latency of proposed protocol is decreased. At last it is validated by experiments that the proposed protocol can guarantee both data transmission latency less than 125 ns and synchronous & correct transmission of wideband data at the same time.

References
[1] Hua S, Li F and Wu L L 2017 Design and implementation of high-speed data transmission interface between boards Radio Engineering 47 (11) 79-82.
[2] Aloisio A, Giordano R, Izzo V and Perrella S 2015 A frequency agile, self-adaptive serial link on Xilinx FPGAs IEEE Transactions on Nuclear Science 62 (3) 955-962.
[3] Sabat S L, Ajay Kumar D and Rangababu P 2009 Reliable high speed data acquisition system using FPGA IEEE 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET) pp 392-396.
[4] Mil D 2014 snapMac: A generic MACPHY architecture enabling flexible MAC design Ad Hoc Networks 17 37-59.
[5] Xilinx 2012 7 Series FPGAs Overview (America: Xilinx Inc.)
[6] Xilinx 2015 7 Series FPGAs GTX/GTH Transceivers User Guide V1. 11 (America: Xilinx Inc.).
[7] Giordano R and Aloisio A 2011 Fixed-latency, multi-gigabit serial links with Xilinx FPGAs IEEE Transactions on Nuclear Science 58 (1) 194-201.