Design of load data simulation source

Yong Sun1,*, Jianmin Wang2 and Jiaxiang Shi3
1Harbin university of science and technology, Harbin, China
2Harbin university of science and technology, Harbin, China
3Harbin university of science and technology, Harbin, China

*Corresponding author e-mail: sunyongmail163@163.com

Abstract. In the remote sensing satellite system, the main function of the data transmission subsystem is to receive all the data transmitted by the payload subsystem and then transmit it accurately to the data receiving station on the ground. In the absence of satellite payloads, to meet the requirements of the joint remote sensing satellite test link, we studied and designed a load data simulation source that can simultaneously output four channels of continuous data at different rates, and there are two operating modes to choose from. This design adopts Kintex UltraScale series FPGA with configurable Microblaze embedded soft core as the core controller, and the peripheral circuit includes four pieces of 2GB DDR4 memory, power supply and LVDS interface output. The actual test results show that the load data simulation source has high stability and good universality, and can perfectly meet the ground debugging and testing requirements of remote sensing satellite data transmission subsystem.

1. Introduction
In the whole structure of remote sensing satellite, data transmission subsystem is an important part. The main function of the data transmission subsystem is to receive the pixel data of the high-resolution image transmitted by the payload subsystem, and then transmit it accurately to the data receiving station on the ground[1]. But payloads are expensive and take a long time to develop. In order to coordinate with the ground equipment for debugging and meet the test requirements, a satellite payload data simulation source was designed and developed to simulate the data and time sequence consistent with the real payload. The simulation source works as shown in figure 1.

Figure 1. working principle of load data simulation source
When the load data simulation source works, it is first plugged into a specific case to connect it to the computer. The original image pixel data generated in advance by the computer is transmitted to the simulated source board card through the bus. According to the corresponding parameters configured by the user, the simulated source board card continuously sends out the pixel data of the original image in a format and sequence meeting the specific requirements.

At present, the following schemes are mainly adopted in the development and design of load data simulation sources at home and abroad:

The first kind is the most methods used by the current, which is PC architecture, his main principle is to develop a special board to the corresponding data flow was simulated, the advantage of this approach is flexible and resolution is higher, defect is simulated data accuracy is not high, unable to realize the output multichannel parallel high-speed data signal [2].

The second is the American COTS system, which combines FPGA and VxWORKs embedded system and can effectively complete the testing of different types of data sources and obtain good results in multi-data source simulation. However, in the 1990s, the hardware performance was low and its application was not popularized[3].

This design adopts high-speed PXIe interface technology and DDR4 storage device, and adopts high-performance Kintex UltraScale series FPGA as the core controller. The design of low bit error rate, large cache capacity and sending multi-channel high-speed parallel data simulation source is realized. Can realize file loading and pseudorandom number generation, with multiple load data simultaneously send function.

2. Function and technical index

The technical indicators of load data simulation source are as follows:

Data output consists of four channels, which are controlled by three groups of clocks. The channels of different clocks are independent from each other and time sequence is independent. Each channel has 6 channels of LVDS signal output, which requires 4 channels to simultaneously send 24 channels of LVDS signal.

Working mode can be selected through pattern configuration. Two modes can be selected to meet different task requirements.

The simulated source uses the file loading method to load the user-specified files into each channel for circular sending, and supports loading multiple files at one time. It is looped from a data channel in sequence according to the file sequence number. Single file size is supported 0-256 MB. The loading speed is no less than 100 MB/s.

3. The project design

The working principle of the load data simulation source proposed in this paper is shown in figure 1. In general, the payload data source board card needs to work in a specific case and communicate with the computer via the bus. The computer transmits the simulated image data to the memory of the simulated source card through the PXIe interface of the case. The simulated source board card then reads the data from the memory into the cache and sends it out at a specific format rate according to the configured parameters.

Considering the design requirements and the development direction of current testing technology, we chose PXIe bus to realize the communication between the data simulation source board card and the computer. FPGA is selected as the core controller of hardware board card design. Labview software is selected as the upper computer software for control.
3.1 The hardware design

The schematic diagram of the hardware design is shown in figure 2.

The entire design of hardware board and card structure consists of four parts:
1) core control part, including FPGA, clock, debugging interface (JTAG) and configuration circuit;
2) the storage part, including the DDR4 memory circuit;
3) isolating chip and DC-DC isolating power supply;
4) LVDS interface circuit.

For FPGA, we chose Xilinx's Kintex UltraScale series, which features the best cost/performance/power ratio. We used the IP core provided by Xilinx company and the IP core we designed and packaged to build our underlying logic.

It mainly includes the following parts:
1) microblaze embedded soft core, which ACTS as the master control function, connects each part to a AXI bus for operation control.
2) In the clock section, there are three input clocks. The 90MHz input clock generates a 100MHz clock through the clocking wizard IP core for use by microblaze. The 150MHz input clock generates three groups of clocks (50MHz, 52MHz, 29MHz) through the clocking wizard IP core, which are used for four output channels (there are two outputs using the same clock). The 300MHz differential clock is used for DDR4.
3) XDMA IP core, used to implement dma transfers, configure our hardware board card ID. According to the configured ID content, the corresponding driver of the data source board card is written, so that our computer case can recognize the board card.
4) DDR4 part, external storage device, used to achieve storage function.
5) Self-built IP core, this part is the unique design of the data simulation source card. This section contains five FIFOs and a control program that outputs the cached data in the FIFO at a specified format rate. The work requirements of the five-channel data are shown in figure 3 and figure 4 below.
As can be seen from the figure, each data transmission route has its own unique effective transmission time and invalid transmission time, and the smv channel also needs to realize its unique controllable number of packets issued during its second effective clock. The data of d, z, n and p channels are one bit wide, and the data of smv channels are two bits wide.

For the smv channel and the p channel, because the two channels are in the choice of output relationship, they use the same channel output, that is, when one channel works, the other channel stops working, so they are both implemented in the same verilog program.

The method to realize the interface logic of five channels is implemented by finite state machine. For smv channel, he needs 7 states: idle state, invalid time state T2, T4, T6, valid time state T1, T3, and T5. The time Settings of these states are configured by the upper computer software through the bus and can be changed at will to meet different requirements. Transitions between states are implemented by state counters. Since the FIFO cache data is eight bits wide and the smv channel is two bits wide, the higher two bits of the eight-bit FIFO cache data will be transmitted to the output when the rising edge of each clock cycle is set in the effective clock state, and the cached data in FIFO will be updated once every four clock cycles. For the valid state T3, in this state we need to realize that we can transfer N packets of data, and the size of N packets is controllable, with 1024 bits of data per packet. Therefore, in this state, when the state counter is full, it is necessary to see how many packets of data have been transferred in the N packet. When the transmitted data is not satisfied, it will jump back to T3 state and continue the transmission. When the data transmission of the N packet is completed, it will jump to T4 state.

Compared with smv channel, p, d, z and n channels are relatively simple. Each channel has only three states: idle state, valid time and invalid time. The implementation is the same as for smv channels.

After the implementation of the five-channel code, it is encapsulated into an IP core. Finally, connect each part according to the requirements.

After completing the timing constraints and pin constraints, run synthesis, run implementation, complete the address mapping on the bus, and assign addresses to DDR, AXI_DMA, etc. Finally completed the underlying hardware design. The resources for the entire project can be viewed in the final project report, as shown in figure 5.

After the FPGA underlying hardware is configured, it is necessary to complete the control of the hardware by the software in the Xilinx SDK. Through the PXIe bus, the simulation data and parameter
configuration on the computer are read and written to the data source board card DDR4 by means of DMA transmission[4], and then the data in DDR4 is written to FIFO, and the various parameter configurations are read and written to the corresponding registers, which completes the preparation before data transmission. The main function is to complete the corresponding operation according to various parameter configuration instructions, such as opening the board card, data transmission and other operations. This completes the hardware and software configuration of the entire FPGA.

3.2 The software design

The main function of the software design is to configure the parameters for the five-channel channel, and to store the prepared image simulation data into the memory of the board card, so that the board card can realize the function according to the instructions. Software design part we want to achieve two parts of the content, respectively is the generation of the DLL and the configuration of the host computer software.

Visual Studio 2015 is the software we choose to generate the DLL. The main function of this part is to realize the control of the underlying hardware and software, so as to enhance the configurability of the whole system. If changes are needed, only the configuration of the software needs to be changed, and the underlying hardware does not need to be changed. What we need to implement in this section is to configure the parameters of the five-channel channel and store the four-channel simulation data source in DDR4 according to a specific sequence and format.

The main contents of the DLL we need to generate include the following functions:

1) Open board card;
2) Close the board card;
3) Initialize the board card;
4) Enable the board card;
5) Reset board card;
6) The d channel is loaded with simulated data;
7) The z channel is loaded with simulated data;
8) The n channel is loaded with simulated data;
9) Simulate data loading of SMV channel or p channel;

Among them, when enabling the board card, the parameters required by each channel will be configured, and the p channel or smv channel will be selected. In this part, we need to generate three files for the following software, namely h file, dll file and lib file.

We use labview2017 to implement the master control software of the upper computer, which is characterized by a graphical operation interface and easy to operate. First, Labview should be used to identify the three files generated in the above process and analyze the vi file of each step. After recognition, the connection should be made according to the board card operation order we need, and the corresponding functions should be realized according to the requirements of each step. The final arrangement result is shown in figure 6:

![Figure 6. vi arrangement and connection](image-url)
4. Testing and analysis

During the test, we selected NI PXI-1045 case of NI company to carry out relevant tests on the load data simulation source board card. After the board card is inserted into the case, the pre-prepared image simulation data will be sent to the board card through the software. The board card will send the data through LVDS interface according to the format and time sequence we set, and then send it to our receiving board card. The receiving board card will feedback the received data back to the computer and test whether our board card will send the data accurately by the compare terminal software.

As shown in figure 7, configure our four-way channel parameters.

![Parameter Configuration](image1.png)

(a) smv, p channel parameter configuration

![Parameter Configuration](image2.png)

(b) d, z and n channel parameter configuration

Figure 7 parameter configuration

After the parameter configuration, the computer sends the image simulation data we prepared in advance for testing to the board card, and the test results are shown in figure 8.

![Test Results](image3.png)

Figure 8 test results
Channels 6, 7, 8 and 9 are the four data channels of our board card. It can be seen that after the continuous loading of 8292 seconds, the four-channel data is loaded normally without any error codes, and the code rate meets our requirements.

After repeated tests, the load data simulation source designed by us can work stably and effectively, and can perfectly realize the signal output of the payload. Therefore, it can perfectly meet the ground debugging and testing requirements of the remote sensing satellite data transmission subsystem.

References
[1] Hong zhao, Xiaoqing Shan, Changyan Xiao. Design and Implementation of data simulation source for baseband data of remote sensing satellite [J]. Computer measurement and control, 2012, 20(2): 411-413.
[2] Wenquan Feng, Xiaolin Zhang. Design and implementation of Compact-PCI-based multi-way telemetry signal data source [J]. Journal of Beijing University of Aeronautics and Astronautics, 2005, 31(9): 1036-1039.
[3] Feifei Liu, Hanbo Cheng, Jingyu Wang. Research and application of scientific Satellite payload EGSE Platform [J]. Journal of Beijing Institute of Technology, 2010, 30(2): 214-219.
[4] Zhaoqing Liu, Weida Du. Design of PXI Express DMA engine based on IP Core [J]. Electronic measurement technique, 2012, 35(7): 43-46.