Design of DDR3 SDRAM read-write controller based on FPGA

Jinhui Yi1,2,3, Mingfu Wang1,2* and Lidong Bai1,2,3

1 Key Laboratory of Science and Technology on Space Optoelectronic Precision Measurement
2 Institute of Optics and Electronics of the Chinese Academy of Science
3 University of Chinese Academy of Sciences Chengdu China

*Corresponding author: Arraby@163.com

Abstract. In order to flexibly adjust the frame delay of real-time image acquisition by high-resolution cameras, which is based on optical fiber communication protocol, and facilitate subsequent control, this article uses MT41J128M16JT-125IT DDR3 SDRAM of Micron company to cache image data. And based on the MIG controller that comes with Xilinx Vivado development tool for continuous read and write control, the results show that when the camera system is designed at 2fps and the system clock is 50Mhz, the system data bandwidth is 2.2Gbps. The selected DDR3 chip has a bandwidth of 6.25Gbps, which can meet the real-time transmission requirements of the design system.

1. Introduction

In recent years, with the rapid development of semiconductor technology, CMOS cameras have also been developing towards miniaturization, low power consumption, and high resolution. In the actual CMOS camera imaging system design, in order to be able to flexibly adjust the time delay of images acquired in real time and facilitate subsequent imaging control, image data needs to be cached. Under the requirements of high speed, large capacity, high bandwidth, and high real-time performance, DDR3 SDRAM uses DDR (double data rate) technology and prefetch technology, which greatly increases the data storage rate and bandwidth [1]. It also has the advantages of small size and low price, so it is the best choice in data storage system design.

This article is based on the MIG controller that comes with Xilinx FPGA to realize the read-write control of DDR3 SDRAM, and finally completed the test and function realization on the Kintex-7 FPGA chip.

2. MIG principle of DDR controller

When reading or writing, after the DDR is powered on, a series of operations such as initialization, ZQ calibration, configuration mode register, and bit width splicing are required [2]. These functions are mainly realized by commands such as active, precharge, refresh, read and write. Obviously it is very unfriendly to users.

Therefore, this design uses the IP core MIG_v4.2 controller that comes with the Xilinx development tool Vivado to read and write the DDR3. The basic principle is shown in Figure 1.
The MIG controller is composed of four parts: user control module, user interface module, memory control and physical layer interface. When using the controller, user only needs to control the read-write logic on the user side, while the IP core will automatically transmit the data, address or control required by DDR3.

First, when the physical layer interface calibration is completed, the signal (init_calib_complete) will be sent immediately. Then user interface module will release the app_rdy signal to tell the user that it is ready to accept instructions. Therefore, only when app_rdy is 1, the read and write control signal app_cmd and address signal app_addr passed to the interface module will take effect. And only when the FIFO for writing data in the IP core is ready, it will release the signal that app_wdf_rdy is high, then the user can write to DDR3 [3-5].

3. DDR3 read and write control
The sequence diagram of read-write commands and read-write address is shown in Figure 2. Take the write operation as an example: only when app_en and app_rdy are both high (ie 1 in the figure), app_cmd (read and write commands) and app_addr (address) are valid.

The write data and the write address signal are not necessarily aligned strictly, and the write data can be delayed by up to 2 clock cycles of the write command (shown in 1 in the figure). The read operation is similar to the write operation.
In the continuous read and write control, the feedback signal sent by the user interface should be fully utilized as a condition for the state machine to jump. And use a counter to control the number of data received. The description of each state is as follows.

S_IDLE: Enter this state when receiving the signal (init_calib_complete) after the physical layer calibration is completed, waiting for the user to send a read (rd_burst_req) or write (wr_burst_req) request.

S_RD: When S_IDLE receives a read request, it enters this state. Only when app_rdy is valid, the address in DDR3 is increased by 8, and the corresponding address counter is increased by 1. At the same time, the number of read data is detected by the data receiving valid signal.

S_RD_WT: As mentioned above, the read data can be several clock cycles later than the read address, so when the address counter reaches the read burst length, the read data still needs to be received. State machine enters the S_RD_WT state to wait for the read data.

S_RD_END: When the read data reaches the burst length, the read operation is completed and jump into S_IDLE.

S_WR: Same as read operation, when S_IDLE receives a writing request, it enters this state. Only when app_rdy is valid, the address in DDR3 is increased by 8, and the corresponding address counter is increased by 1. At the same time, the number of written data is detected by the data acceptance signal. But unlike the read operation, the MIG controller will first store the data into the FIFO when writing data, and the writing of data generally precedes the address.

S_WR_WT: Waiting for the data to be stored in the corresponding address.

S_WR_END: After the write operation is complete, jump into S_IDLE.

4. DDR3 read and write test
The design uses the MT41J128M16JT-125IT DDR3 SDRAM of Mircon company. The internal storage has 8 banks, the data width is 16bit, and 16M memory cell; 14bit row address line and 10bit column address line. So for the MIG controller, it should have a 27bits address signal. The maximum IO clock
frequency supported by the chip is 800Mhz (that is, the frequency of DDR3), which can well meet the system data bandwidth.

In the read-write test for DDR3, read first and write later, and in order to facilitate data capture, read and write to the same address repeatedly. Table 1 shows the resource utilization of the DDR3 read-write controller after the final implementation. It can be seen that the entire read-write module has simple control and low resource utilization, which is convenient for code migration.

| Resource | Utilization | Available | Utilization % |
|----------|-------------|-----------|---------------|
| LUT      | 9538        | 203800    | 4.68          |
| FF       | 9484        | 407600    | 2.33          |
| BRAM     | 11          | 445       | 2.47          |
| IO       | 48          | 500       | 9.60          |
| MMCM     | 3           | 10        | 30            |

Figure 5 shows the written data captured in real time by the integrated logic analyzer ILA (Integrated Logic Analyzer) that comes with Vivado. It can be seen that the read-write controller has successfully written all 512 numbers. In order to further verify the correctness, it requires reading the whole stored data.

Figure 5. The test results of DDR3 data writing

Figure 6 indicates the continuous reading of the data just stored in the address, and you can also see the nice correspondence between the stored number and the address.

Figure 6. The test results of DDR3 data reading

5. Conclusions
In order to facilitate the storage of a large amount of pixel information, and meet the purpose of frame delay for high resolution cameras. This article proposes using FPGA as the control chip and DDR3 as the storage device to complete the corresponding design. When the camera system is designed for 2fps and the system clock is 50Mhz, the data bandwidth of the system will reach 2.2Gbps. The internal pixel processing bandwidth is 64bit, so FPGA processing bandwidth is 3.2Gbps, and the selected DDR3 chip
bandwidth is 6.25Gbps, which can meet the real-time transmission requirements of the design system. Finally the reading and writing test shows that the DDR3 controller module is easy to operate and has low resource utilization, and it can be easily connected with the camera readout module.

References

[1] Ding, P.P., Zheng, L., Li, H. (2018) Design and implementation method of DDR3 SDRAM controller based on FPGA. Information Recording Materials, 19(05): 64-66.

[2] Wang, M.F., Yang, S.H. (2009) Design of SDRAM controller in the real-time display system of large-area CCD images. Computer Applications, 29(05): 1449-1451.

[3] Song, M., Zhao, Y.X., Lin, Q.Q. (2016) Design and optimization of DDR3 SDRAM controller based on FPGA. Electronic Science and Technology, 29(11): 47-50.

[4] Zong, K. (2017) FPGA-based DDR3 controller design. Electronic Measurement Technology, 40(01): 118-122.

[5] Ding, N, Ma, Y.C., Q,L., Han, S.. (2014) User interface design of DDR3-SDRAM controller based on FPGA[J]. Science Technology and Engineering, 14(17): 225-229.