

## Design of a New Data Storage System Based on Serial Communication

Zhang Bo and Zhang Gang

Department of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China

**Abstract:** In order to improve the accessing capacity for CPU, we design a new storage system based on serial communication. The system utilizes some pieces of DDR SDRAM as storage space. A new serial communication protocol and data frame format is introduced in this study. With the serial communication protocol, the CPU will access the data storage with only two unidirectional serial buses. The storage system has the advantages of easy expansion of capacity, simple structure and flexible application. The whole design is realized in VHDL and verified on Xilinx FPGA development platform EDK.

**Keywords:** Data storage, DDR SDRAM, EDK serial communication

### INTRODUCTION

DDR SDRAM memory is widely used in data storage system with the advantage of large storage capacity, fast speed and low price. Expansion of storage capacity is the key technology to improve the CPU system performance (Qin-sheng *et al.*, 2010; Jing *et al.*, 2008). However, DDR SDRAM has larger number of pins and will occupy many IO ports for CPU processor. The CPU processor will not connect multiple DDR SDRAM and decreases the accessing storage capacity, so the performance of CPU will not be fully displayed.

The study presents a new storage system based on serial communication. The system uses the FPGA as the system controller and connects multiple DDR SDRAM (Xiao-Ling *et al.*, 2011; Xiao-CHEN and Min-Xuan, 2011). We define a serial communication protocol and the data frame format. The CPU processor will access any DDR SDRAM by two serial communication buses, which will expand the accessing capacity for CPU and improve the performance of the whole system greatly.

**Principle of the system:** System structure is shown in Fig. 1. The CPU connects the FPGA by two unidirectional serial buses. The CPU transfers the data to FPGA with the red serial bus. By analyzing the received data, FPGA extracts the valid data and writes the data to DDR SDRAM specified by the CPU. FPGA reads the data from DDR SDRAM and packets the data to frame format. Then the data is transferred to CPU by the blue serial bus. Therefore the CPU will access any DDR SDRAM easily.

### SYSTEM DESIGN

The interior structure of the system is shown in Fig. 2. The red unidirectional serial bus is exclusive to

CPU and is used only by CPU to transfer data. The blue unidirectional serial bus is shared by the memory controllers and the memory controllers transfer data to CPU with the same bus.

The CPU sends data to DDR SDRAM by the red serial bus. When the bus interface module receives the data, the bus interface module extracts the destination address from the data and compares the address with its node address. If the two addresses are the same, the bus interface module receives the data. Otherwise it will abort the data.

The bus interface module analyzes the received data frame and extracts the command (reading or writing), Bank address, row address, column address and the data length. If the command is writing, the bus interface module transfers the above signals to DDR SDRAM controller and the data is written to DDR SDRAM. If the command is reading, the DDR SDRAM controller reads the data from DDR SDRAM and sends the data to bus interface module. The bus interface module packets the data to serial data frame format and sends the data frame to CPU by the blue serial bus.

**Unidirectional serial bus data frame format:** The data frame of unidirectional serial bus concludes the data transferred from CPU to bus interface module and the data sent from bus interface module to CPU. The data frame format from CPU is shown in Fig. 3 (Fuyong *et al.*, 2011).

DA means destination address. It consists of unicast address and multicast address. The range of unicast address is 0x00 to 0xFE and the number means the specified address for every memory node. The multicast address is 0xFF and CPU sends data to all memory nodes. The length of DA is one byte and the CPU will access as many as 255 DDR SDRAM. The command field represents the CPU's command for

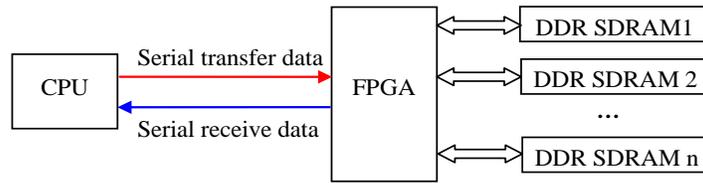


Fig. 1: System structure

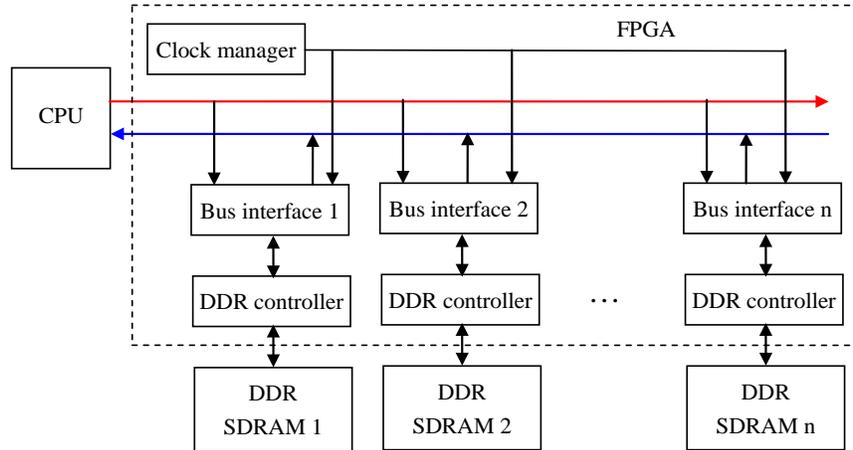


Fig. 2: System interior structure

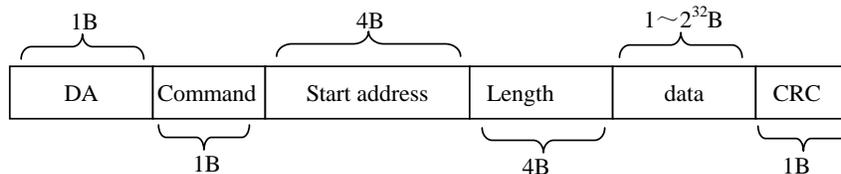


Fig. 3: Data frame format from CPU

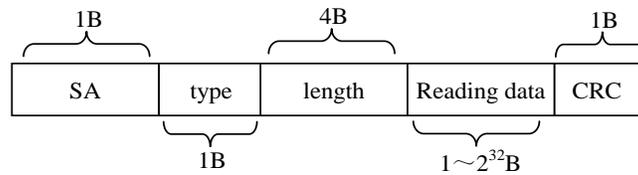


Fig. 4: Data frame format from memory node to CPU

DDR SDRAM and it includes writing and reading. The length of this field is one byte and 0x00 means writing command and 0xAA means reading command. The start address field means the starting address for CPU to access DDR SDRAM. The length of this field is four bytes and the 31 to 30 bit is the Bank address, the 29 to 16 is the row address and the rest is column address. The data length field means the number of data for writing to or reading from DDR SDRAM. The length of

this field is 4 bytes. The data field is the writing data for DDR SDRAM from CPU and its length is 1 to 232 bytes. The CRC field is the verification field and its length is one byte.

The data frame format from memory node to CPU is shown in Fig. 4. SA means source address and it is the address of every memory node. The length of this field is one byte. The type field represents reading data or checking error and 0xAA is data reading and

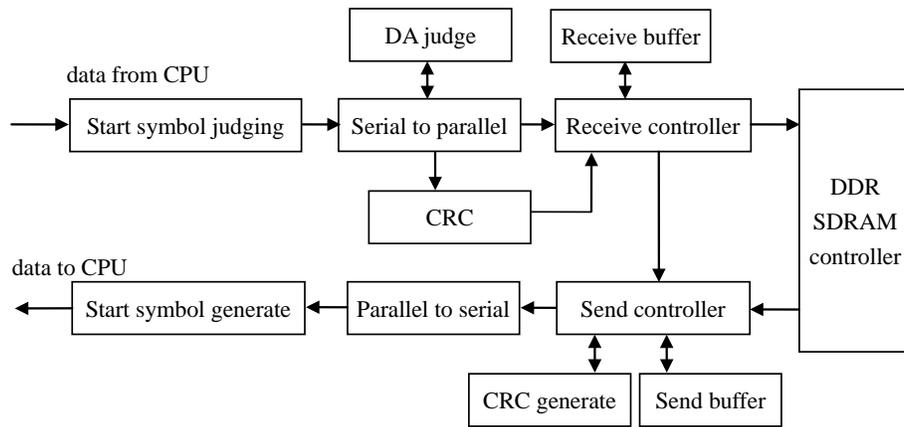


Fig. 5: Interior module of bus interface module

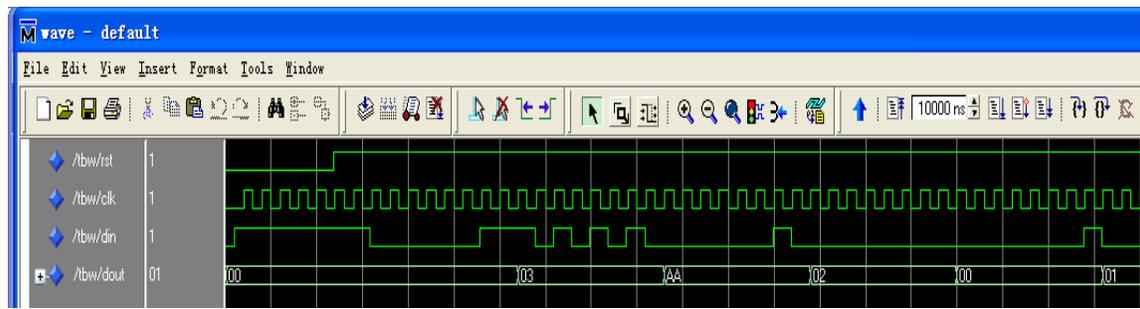


Fig. 6: Simulation waveform of the serial to parallel module

0xF0 means data checking error. If the data checking is error, the length field and the reading field are 0x00. If the data checking is right, the length field represents the number of reading data and the length of this field is 4 bytes. The reading data field is the number of data reading from DDR SDRAM and its length is 1 to 232 bytes. The CRC field is the data checking value with the CRC8 algorithm and the length of this field is one byte.

**Clock manage module:** The CPU sends data to memory nodes through unidirectional serial bus. Since there is no clock and enable signal, the bus interface module receives the data only according to the baud rate. The baud rate is set according to the performance parameter of CPU and the clock manage module will generate the appropriate clock by the DCM IP in FPGA. The clock is sent to every bus interface module. The bus interface module receives the data from CPU or sends data to CPU with the synchronous of the clock.

**Bus interface module:** The bus interface module receives the data from CPU and unpacks the data. If the destination address is the same with the address of this bus interface, the bus interface module receives the data; otherwise it will abort the data. If the destination

address is multicast address 0xFF, all bus interface modules will receives the data.

The bus interface module packets the data reading from DDR SDRAM and sends the packet data to CPU by the unidirectional serial bus. The interior structure of this module is shown in Fig. 5.

**Start symbol judging module:** The start symbol judging module is used to judge when the CPU sends data. The unidirectional serial bus is in high level when the bus is in idle state. The bus is driven low when CPU is ready to send data. Then CPU sends data with the baud rate. If the start symbol judging module detects the serial bus transferred from high to low level, it begins to receive the data with the baud rate.

**Serial to parallel module:** The module converts the serial data received from the unidirectional serial bus to 8 bits parallel data. Since the serial data transferring using the low bit in the former, the module applies 8-bit right shift register and gets the serial data right shift. When receive 8 serial data, the register outputs 8-bit parallel data. The simulation waveform of this module is shown in Fig. 6.

As shown in Fig. 6, rst is the system reset signal and low level is valid. Clk is the sampling clock signal

generated from the clock manage module. Din is the serial input data. Dout is 8-bit parallel output data.

**Destination address judging module:** The module receives and judges the first byte output from the serial to parallel module, which is the destination address. If the address is less than 0xFF, the data frame is the unicast frame. If the destination of the data frame is the same with node address of the bus interface module, the data from is send to this bus interface module, otherwise, the bus interface module abort this data frame. If the destination address is 0xFF, which is the multicast address, then all bus interface modules will receive the data frame.

**Receive control module:** The module receives the parallel data output from the serial to parallel module. According to the data frame format, the module extracts the command signal, Bank address, row address, column address, data length and writes the valid data to receive buffer. Then the command signal, Bank address, row address and valid data are sent to the DDR SDRAM controller.

**CRC checking module:** The module receives the parallel data output from the serial to parallel module and verifies the parallel data with the CRC 8 algorithm (Zhan-Kun *et al.*, 2007), whose formula is  $x^8+x^5+x^4+1$ . The CRC checking value generated from the module is compared with the CRC output from the serial to parallel module. If the two values are the same, the data frame is received correctly; otherwise, it will abort the data frame and sends the checking failure signal to CPU.

**Sending control module:** The module receives the data sent from the DDR SDRAM controller and sends the data to send buffer. Then the data is packet to data frame format. The data is sent to parallel to serial module with the baud rate clock and the data is also sent to CRC checking module for generate the checking value.

**Parallel to serial module:** The module changes the 8-bit parallel data from the sending control module to serial data and transmits the serial data with the frequency of the baud rate.

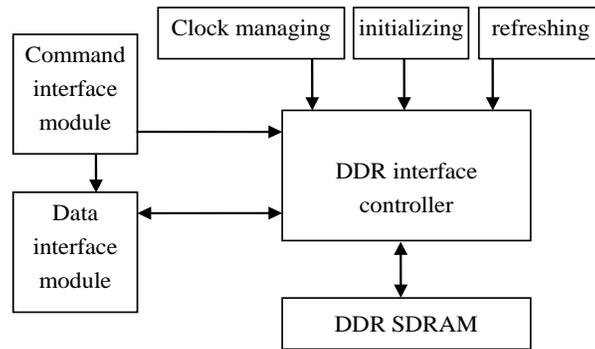


Fig. 7: Interior structure of DDR SDRAM

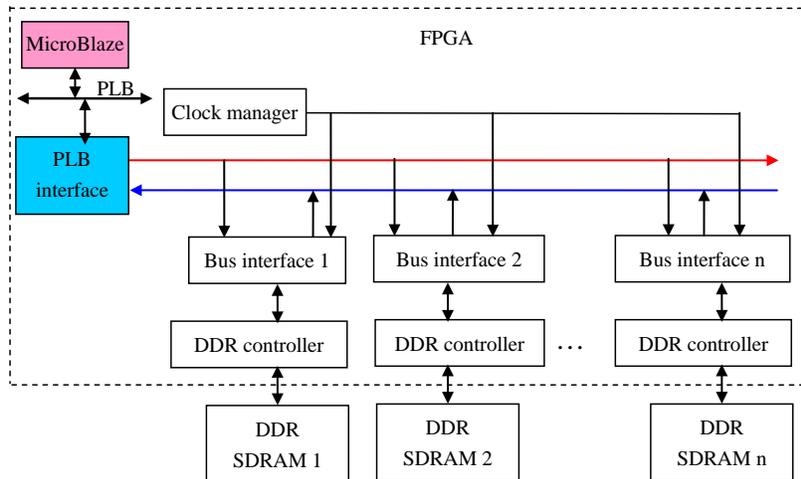


Fig. 8: System verification module

**Start symbol generating module:** Start symbol generating module gets the serial bus to low level before it send the serial data. Then it sends the serial data to the unidirectional serial bus. When the data is sent over, the serial bus is driven high resistance state.

**DDR SDRAM controller module:** The module receives the reading or writing command, address, data and data length information, which from the bus interface module. Then it accesses the DDR SDRAM memory. The interior structure of this module is shown in Fig. 7 (Xiao-Tong *et al.*, 2007; Yang and Zheng-Hui, 2006).

Initializing module initializes the DDR SDRAM and sets the burst length, burst type, CAS latency. Since DDR SDRAM is dynamic memory, it needs refreshing operation; otherwise, the content of the memory will be lost. The time between 2 refreshing command is the average time for refresh all rows in 64 ms. If the row address is 12 bit, the refreshing command time is  $64 \text{ ms}/2^{12} = 15.625 \text{ us}$ . The clock managing module generates the clock signal and inverted clock signal to access the DDR SDRAM. It also generates the double clock signal to send or receive data on the rising or falling edge of the double clock signal.

The command interface module receives the reading and writing command, accessing address and data length signals. It generates the ACTIVE, read, write and address command and sends them to DDR interface controlling module. It also generates reading enable and writing enable signal to data interface module to send or receive data.

The data interface module receives the data sent from the bus interface module. When the writing enable signal is in valid, the data is sent to DDR interface module. When the reading enable signal is in valid, it receives the data which is from DDR interface module. According to the received command signal, DDR interface module generates the RAS, CAS and WE signal to control DDR SDRAM.

## SYSTEM VERIFICATION

Utilizing the Micro Blaze soft processor core in Xilinx FPGA as the system CPU, we design a system verification platform with the EDK platform (Tian and Wen-Xin, 2010). The system verification module is shown in Fig. 8.

In EDK development platform, the PLB interface module is used as the user IP core and is linked to the PLB interface. The Micro Blaze processor will access the PLB interface module directly. The parallel data coming from the CPU is packet to serial data frame by the PLB interface module and the serial data is sent to the unidirectional serial bus. Then the bus interface module receives the serial data and judges to receive the data or not by the destination address in the data frame. If the bus interface module receives the reading command, the DDR SDRAM controller module will

access the DDR SDRAM and the data reading from DDR SDRAM is packed to serial data. The serial data is sent to PLB interface module by the unidirectional serial bus and the PLB interface module will get the serial data to parallel data and send it to CPU by PLB bus. The Micro Blaze processor will access any DDR SDRAM memory, which will enhance the access storage for Micro Blaze.

## CONCLUSION

In this study, we design a new storage system using DDR SDRAM and define a serial transmission protocol for the system. The CPU will access the storage system by two unidirectional serial buses. The system will enhance the accessing memory for the CPU, which will improve the system performance. The whole design is described in VHDL and verified in Xilinx FPGA platform.

## ACKNOWLEDGMENT

This study is supported by the Chinese national natural science fund project. 60772101.

## REFERENCES

- Fuyong, L., L. Yong-Xin and B. Xiong-Zhu, 2011. Simplified serial bus based on time-triggered architecture. *J. Nanjing Univ. Aeronaut. Astronaut.*, 43(4): 544-549.
- Jing, W., Y. Junkai and W. Xue-Wei, 2008. Shared storage architecture for parallel database. *J. Comput. Inform. Syst.*, 4(1): 375-382.
- Qin-Sheng, M, C. Yang, Y Jun and Z. Ning, 2010. Research on the multiport memory controller IP core. *J. Xidian Univ.*, 37(1): 142-147.
- Tian, J. and L. Wen-Xin, 2010. SOPC Design based on ISE and EDK tools. *Microcomput. Inform.*, 26(10-2): 181-182.
- Xiao-Chen, G. and Z. Min-Xuan, 2011. Design and implementation of a fpga based gaussian random number generator. *Chinese J. Comput.*, 34(1): 165-173.
- Xiao-Ling, M., H. Wen-Jun and J. Jian-Xiang, 2011. Designed and implementation of FPGA based high-speed bus for control system. *J. Zhejiang Univ. Eng. Sci.*, 45(11): 2043-2049.
- Xiao-Tong, Z., W. Jing-Cun and W. Qin, 2007. High-speed network access technology based on DDR memory bus. *J. Univ. Sci. Technol. Beijing*, 29(11): 1158-1162.
- Yang, L. and L. Zheng-Hui, 2006. Design of DDR SDRAM controller in video decoder. *Comput. Eng.*, 32(1): 240-241.
- Zhan-Kun, B., Z. Yi-Meng and H. Zhi-Ping, 2007. Study on CRC parallel algorithm and its implementation in FPGA. *Chinese J. Sci. Instr.*, 28(12): 2244-2249.