Design of A Multi-channel High Speed FIFO Applied to HDLC Processor Based on PCI Bus[1]

Bin Huang[2], Zhigong Wang, LuFeng Qiao, Yuanlin Lu

Institute of RF & OE ICs, Southeast University, Nanjing 210096, China

Abstract: In this paper, a design for a multi-channel high speed FIFO (First-in First-out) is presented. We know FIFO is widely used in various fields of data processing. Especially in the chip of high speed operation access, FIFO is a key device. This paper describes in detail data structure, algorithm and design method of the FIFO that is to support 128 logical channels and throughput maximum of 150Mbps. The FIFO's important feature is its structure of data buffer manager. The FIFO succeeds in functional simulation and timing verification on FPGA (Field Programmable Gate Array). Because the FIFO is applied to high speed HDLC based on PCI, its function is also tested successfully through FPGA under environment of a real-time operation system—VxWorks.

Keywords: FIFO, HDLC, Multi-channel, PCI

I. Introduction

In developing age of digital communications, higher bandwidth and higher speed is required for multimedia integrated services data network with transport of voice, video, image and data. Therefore all kinds of networks and data switch devices complying with HDLC protocol established by ISO (International Standard Organization), such as accessing devices of wide area network, frame relay system, XDSL, ISDN and routers, use HDLC processor to carry out the functions of communication and control in central system. The FIFO in application to HDLC processor implements data management and data alternation. In recent years, the data processing chip, which has much higher speed and much more capacity, is needed urgently. Multi-channel high speed FIFO plays an important part in the chip, so design of the FIFO module becomes one of some core technology of the chip.

Multi-channel high speed FIFO is designed in order to adapt well to data processing's demand for multi-user and multi-operation. The kind of FIFO not only increases interface rate of data and the number of logical channel and improves data structure and data algorithm so as to relieve host system's burden and optimize performance of the whole the chip, but also meets the requirement of POS (Packet Over SDH) system. Accordingly, the FIFO has been applied universally to a great many networks and some interface devices that are consistent with SDH (Synchronous Digital Hierarchy) standard.

Multi-channel high speed FIFO is separated from two independent modules, which consist of data buffer RAM (DBRAM), channel status RAM (CSRAM), block pointer RAM (BPRAM) and transport control unit.

II. Features of Multi-channel High Speed FIFO

The block diagram of Multi-channel high speed FIFO is shown in figure1. As data buffer manager, FIFOs that are First-in First-out stack have the same data structure as RAM in general, but their data access and data management are different each other. These FIFOs applied to various fields have their own features and optimization in order to meet requirements of some specific fields. However, the multi-channel high speed FIFO is put into use in HDLC processor, so its features are shown below:

- Supports to data buffer management for E1 link 2Mbps, single channel 34Mbps and 8 parallel physical interfaces 155Mbps
- Supports up to 128 logical bi-directional HDLC channels and up to data throughput maximum of 150Mbps
- Multi-channel high speed FIFO is organized into blocks to manage data
- Supports to contain 8K bytes storage space in both receive and transmit direction, which maximize PCI bus efficiency
- A block size of 4 dwords allows maximum flexibility
- Supports to be initially configurable channel status RAM and block pointer RAM by host system

III. Data Structure of Multi-channel High Speed FIFO

The data structure of multi-channel high speed FIFO is important element and core of the whole FIFO module controlling data buffer management.

It is shown in figure 2 that DBRAM composed of 512 blocks contains 8Kbyte (512×16) storage space and a

1. Founded by the Office of Science and Technology of Jiangsu Province

2. Corresponding author, Email: huangbin@seu.edu.cn

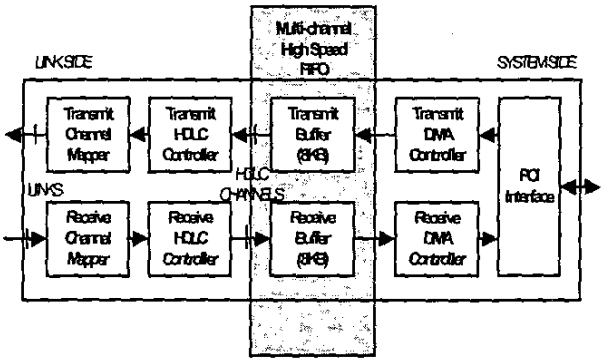

Fig. 1 Block Diagram of Multi-channel High Speed FIFO

block is defined as four dwords (i.e. 16 bytes). In fact, DBRAM is composed of 512 FIFOs, data depth of which is 16 bytes and data width of which is the same each other. A block is organized into a FIFO, and several blocks are link-listed together to form a chain. Therefore on the one hand the FIFO's design is simplified, on the other hand its design become more difficult due to increasing BPRAM included in the FIFO module.

Figure 2 shows BPRAM's address space corresponds to 512 blocks. Each address unit stores a block pointer. Blocks are link-listed together to form a block chain, which also corresponds with a logical channel. Hence a part of DBRAM corresponding to the block chain creates a channel FIFO to work for data buffer of the logical channel, and 128 logical channels have 128 channel FIFOs. These channel FIFOs, which have the minimum size of 4 blocks up to the maximum size of 512 blocks, are assigned by software configuration of host system. In brief, the multi-channel high speed FIFO can be thought of as lots of channel FIFOs, which take on uniform data width as well as different data depth and perform simultaneously data buffer management for a variety of different channel.

Staring block pointer of channel FIFO in DBRAM is given by CSRAM. Each block in BPRAM points to the next block in block chain, and the last block in block chain points to staring block pointer. The block chain operates round to perform data buffer management with increasing of the written data.

IV. Receive Module and Transmit module of Multi-channel High Speed FIFO

The multi-channel high speed FIFO contains receive module in receive path and transmit module in transmipath module comprises DBRAM, CSRAM and BPRAM. Figure 3 illustrates logical diagram of the FIFO. Every fixed storage space is assigned to each of logical channels, storing initial configuration information and instantaneous status information. Initial configuration information includes Staring block pointer, block depth, receive high water mark, transmit low water and so on. Instantaneous status information includes reading block pointer, writing block pointer, the current number of block, pointer offset in a block, readable flag and writable flag etc.

The high water mark indicates to the device how many Blocks should be written into the receive FIFO by the HDLC controllers before the DMA will begin sending the data to the PCI Bus. When the DMA begins reading the data from the FIFO, it will read all available data and try to completely empty the FIFO even if one or more EOF (End Of Frames) are detected.

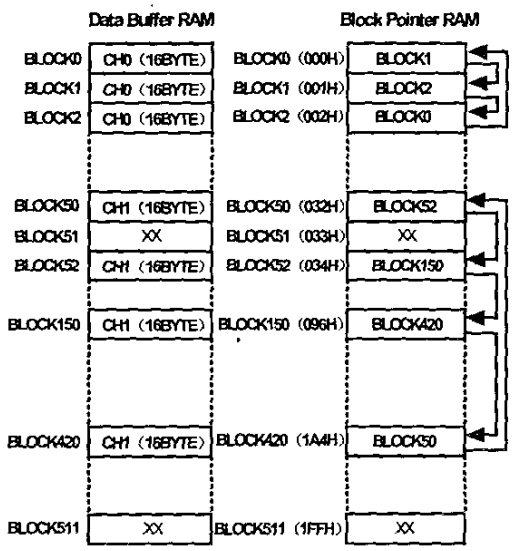

Fig. 2 Data Buffer Structure

The low water mark indicates to the device how many Blocks should be left in the FIFO before the DMA should begin getting more data from the PCI Bus. When the DMA begins reading the data from the PCI Bus, it will read all available data and try to completely fill the FIFO even if one or more EOF are detected.

CSRAM's initial status information is defined by host system, and instantaneous status information is dynamic. After a logical channel finish reading or writing, the storage space corresponding with the logical channel is refreshed in CSRAM to store the newest status information of the current channel. Whenever the new data need reading or writing, the last time statue information is read from CSRAM to be reloaded into the state machine for processing the new data.

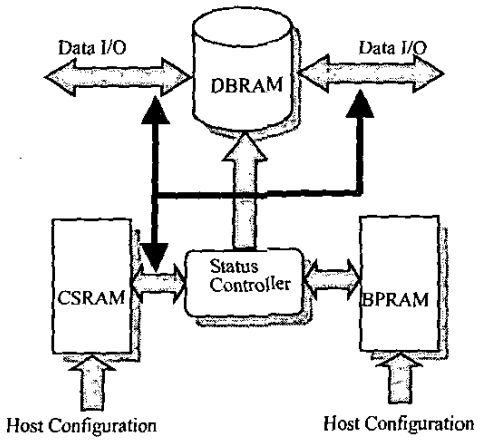

Fig. 3 Logical Diagram of Multi-channel High Speed FIFO

A. FIFO Receive Module

In receive direction, FIFO receive module is divided into three sections: write state machine, read state machine and control state machine. Write state machine writes HDLC data into a channel FIFO in DBRAM. Read state machine transfers a channel FIFO data from DBRAM to receive DMA controller. Control state machine is responsible for CSRAM's reading and writing, BPRAM's reading and writing, application queue for receive DMA controller. Figure 4 shows flow chart of receive FIFO module.

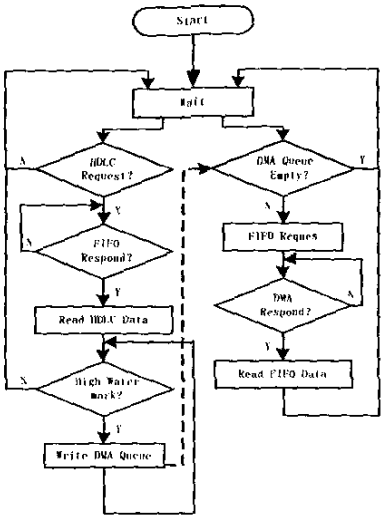

Fig. 4 Flow Chart of Receive FIFO Module

B. FIFO Transmit Module

In transmit direction, FIFO transmit module is divided into four sections: write state machine, read state machine,

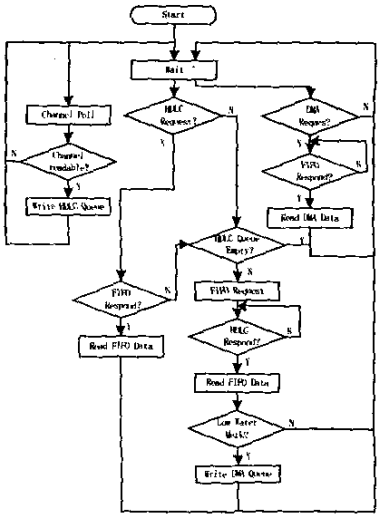

Fig. 5 Flow Chart of Transmit FIFO Module

control state machine and channel poll state machine. Write state machine writes data transmitted by DMA controller into a channel FIFO in DBRAM. Read state machine transfers a channel FIFO data from DBRAM to HDLC transmit processor and controls application queue for transmit DMA controller. Control state machine performs the same functions as it in receive path. Channel poll state machine is responsible for application queue for HDLC transmit processor.

Flow chat of transmit FIFO module is shown in figure 5. We can see that transmit FIFO module is different from receive FIFO module and is more complex than receive module. Firstly, the FIFO module has two application queues, which are considered to be the need of data buffer and transport delay between channel FIFO and HDLC processor, or DMA controller; Secondly, starting state control is not the same as state control of other time segment. In the beginning, all channel FIFOs have no data, and two queues are also empty. Therefore DMA controller makes a request for channel FIFO forwardly. After this, as long as the application queue for transmit DMA controller is not empty, it will request DMA controller to transfer data from DMA controller to the channel FIFO. Thirdly, channel FIFO will be requested suddenly by HDLC processor to transmit data to HDLC processor sometime.

V. Functional Simulation and Timing Verification

The concept of TOP-DOWN is throughout the design of multi-channel high speed FIFO. As hardware description language, Verilog HDL language is used to become the design input and to carry out language compile, logic synthesis and optimization, functional simulation, placing and routing, timing verification in Xilinx Foundation3.1

environment. We utilize Xilinx company's FPGA-Sparta II X2S200 to implement successfully hardware functional verification on the platform of real time operation system. We use two logical channels to show several key points of simulation waveforms in both receive and transmit directions.

A. Simulation Waveform of FIFO receive Module

Channel 0 is defined, depth of which is 3 blocks and receive high water mark of which is 2 blocks. In figure 6 and in figure 7, when HDLC processor sends a request signal of channel 0 and 'channel_req' is set to 1, the channel 0 FIFO is written into 2 blocks. At 7us, 'request_to_dma' is set to 1, which indicates that the size of channel 0 has reached high water mark, and meantime the channel 0 makes a request for DMA controller. As 'dma_read' is set to 0, DMA controller reads out 2 blocks written by HDLC processor continuously in the channel 0 FIFO until the channel 0 FIFO become empty and 'block_count' is set to 0. Synchronization signal 'rfifo_read_ready' is set to 1 at the same time.

DBRAM is made up of dual RAM in FPGA, so figure 7 shows that between 7.2us and 8.2us, when DMA controller is reading data of the channel 0 FIFO, HDLC processor allows to sequentially write channel 0's data F000000000 and channel 1's data 0000000001 into channel FIFO.

B. Simulation Waveform of FIFO transmit Module

In figure 8 and figure 9, channel 1's depth is 2 blocks and its low high mark is one block. In the beginning, application queue for DMA is empty. At 2.86us, when 'write_from_dma' is set to 1, DMA controller writes data into the channel 1 FIFO until the channel 1 FIFO is filled fully. That 'read_allow' is set to 1 indicates the channel 1 FIFO is readable at present. The channel poll state machine is responsible for writing channel NO. into application queue for HDLC. When ' tfifo_req' is set to 1 at 5.08us, HDLC processor will read out a dword 0000000001 from the channel 1 FIFO after it responds. Because it makes a burst request for the channel 1 FIFO between 5.6us and 7.08us, HDLC processor takes three dwords again as following: 0000000002, 0000000003 and 0000000004. Now HDLC processor has read out one block from the channel 1 FIFO, so low water mark is reached. Both the channel NO. and some channel information can be written into application queue for DMA controller. Meantime the channel 1 FIFO is still readable, so it can be written into application queue for HDLC once more by the channel poll state machine and requests HDLC processor again. We can see in figure 10 that when HDLC processor is reading out the other block from the

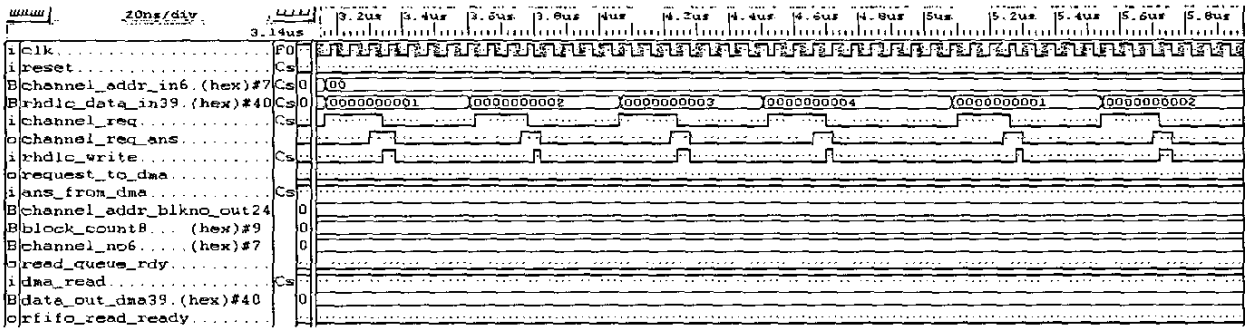

Fig. 6 Simulation Waveform of FIFO Receive Module 1

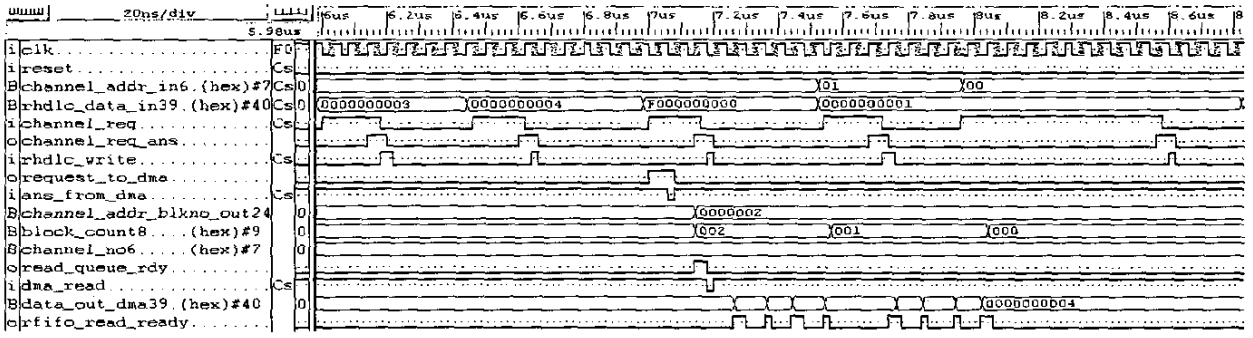

Fig. 7 Simulation Waveform of FIFO Receive Module 2

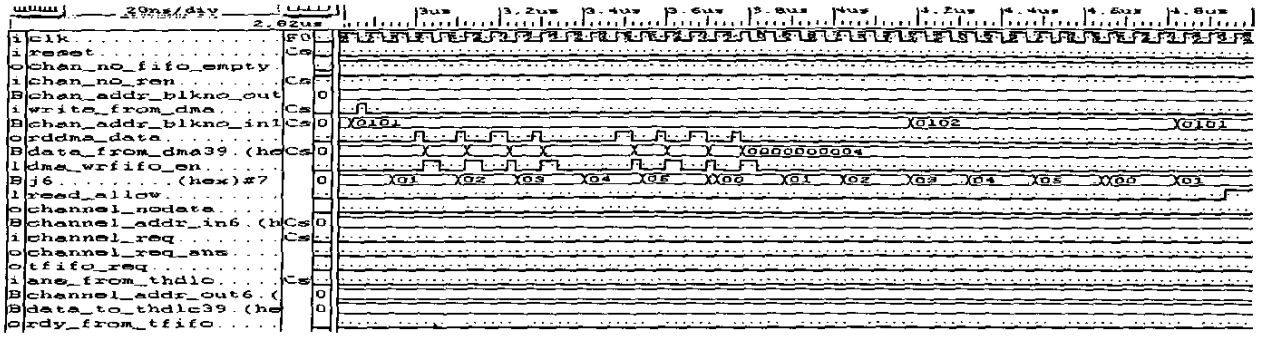

Fig.8 Simulation Waveform of FIFO Transmit Module 1

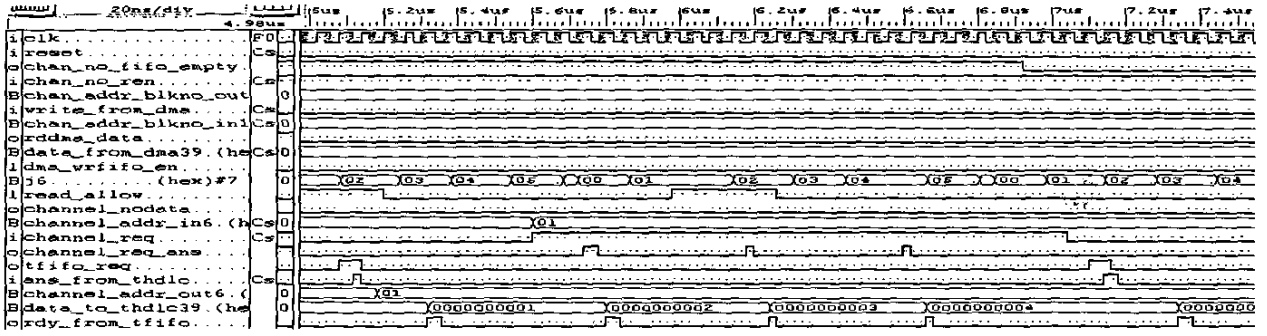

Fig.9 Simulation Waveform of FIFO Transmit Module 2

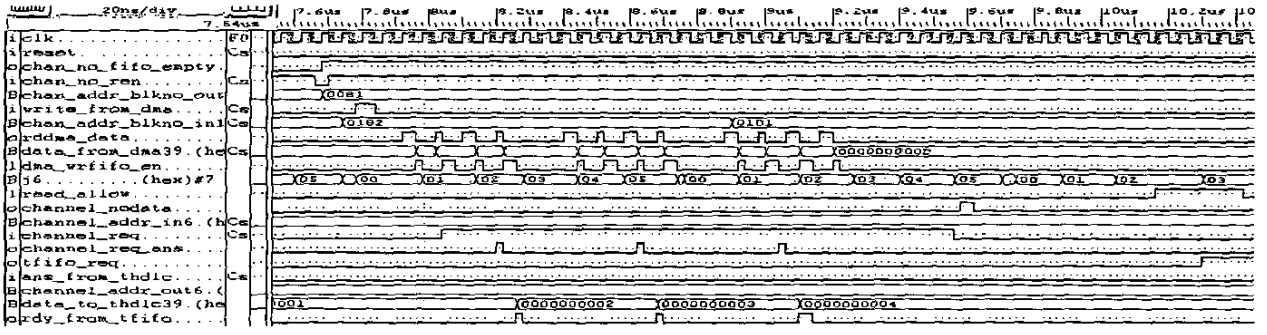

Fig.10 Simulation Waveform of FIFO Transmit Module 3

channel 1 FIFO between 7.04us and 9.52us, 'write_from_dma' is set to 1 at 7.8us and DMA controller writes three blocks into channel 2 FIFO. According to simulation waveforms above, we conclude that two modules work well and timing of them is correct.

VI. Conclusion

Multi-channel high speed FIFO is a key module in the project of "Multi-channel high speed HDLC link manager based on PCI Bus", which is a scientific project of Jiangsu province. This paper studies a logical design of the multi-channel high speed FIFO on FPGA. We have implemented functional verification successfully. The results of simulation waveforms show that the presented design gives a good performance to meet well targets of our design.

References

[1] Yuwen Xia, "Verilog HDL Design Technology of Complex Digital Circuit and system", Beijing University of Aeronautics and Astronautics, August 1998.

[2] Xuan Jiang, Chunhua Zang, "Digital System Design and PLD Application Design", Publishing House of Electronics Industry, pp. 194-196, January, 2001

[3] DS3131 (Bit Synchronous HDLC Controller) Preliminary Data Sheet Version 2, Dallas Semiconductors Inc., pp. 8-9 66-68 January 1999.