International Journal of Combined Research & Development (IJCRD)
eISSN: 2321-225X; pISSN: 2321-2241 Volume: 1; Issue: 8; December–2013

DESIGN OF HIGH SPEED LINK BFFERB FOR HDLC PROCESSOR

Mr. Mohan S R
Asst. Professor, Department of EEE,
PESIT Bangalore South Campus,
Electronics City, Hosur Road ,Bangalore,
Karnataka , India -560068
Email:mohan.sr43@gmail.com

Abstract- **In this paper "Design of High Speed Link Buffer for HDLC Processor" we designed a High Speed FIFO that supports multi-channel HDLC. In the age when digital communications technology is experiencing an unprecedented growth higher bandwidth and better speed is the key requirement for transport of multimedia content (voice, video, image and data) over integrated services data network. Therefore all types of networks complying with HDLC protocol use HDLC processor to carry out communication and control functions. In our proposed system High speed high bandwidth communication needs high speed access and processing of data. FIFO is widely used in the field of data processing. Especially FIFO is a key device in the operation of high speed chips The important feature of FIFO is its data buffer manager. The structure, algorithm of FIFO is designed to support up to 128 logical channels and throughput of maximum 150 mbps. The FIFO in application to HDLC processor implements data management. This kind of FIFO not only increases the interface rate of data and the number of logical channel, but also improves data structure and data algorithm so as to relieve the host system's burden and optimize performance of the whole chip.**

Keywords - FIFO, Multichannel, PCI, HDLC.

I. INTRODUCTION:

A High Speed Link Buffer also called a multichannel First in First out (FIFO) memory is a data-buffering aid that helps to cater to the ever increasing demands for higher bandwidth and higher speeds for multimedia integrated services data network, supporting the transport of voice, image and data. The FIFO uses the principle of queue processing technique for servicing conflicting demands by exhibiting a First Come First Served behavior. The key feature of the FIFO is its structure of data buffer manager. It is composed of a large number of interconnected buffering Elements called channels working in parallel to meet conflicting demands of serving in different time domains. FIFO is often used to safely pass data from one clock domain to another asynchronous clock domain. Thus FIFO memories are used to couple subsystems with different data transfer rates. An asynchronous FIFO is one which data values are written sequentially into a FIFO buffer using one clock domain and data values are sequentially read from the same FIFO buffer using another clock domain, where two clock domains are asynchronous to each other.[7] FIFOs can be designed and built for a specific application, such as digital signal processing, printing, networking systems, routers, frame relay switches and Telecommunications. In this era of digital communication all kinds of networks such as wide area network, XDSL,ISDN complying with HDLC protocol use HDLC processor to manage control and communication functions in the central system. The FIFO in application to HDLC processor facilitates data management by increasing interface rate of data and number of logical channels and throughput. A High Speed Link Buffer for HDLC processor can be realized by designing an efficient multi-channel FIFO using SRAM in an FPGA chip.[3]

II. STATEMENT OF THE PROBLEM

In the Information Age, digitized data such as Speech, music, video, text, pictures, etc is stored, processed, and transmitted on communication links according to a digital format so that it can be readily recognized and processed by a computer system. Moreover this data is highly bursty in nature. Additionally, the modern computer systems cater to many devices at the same time where CPU capacity such as its bandwidth, memory is shared amongst many devices attached to it not only that the systems and transmission mediums that operate at different clock speeds but they observe various communication protocols such as HDLC protocol. These communication protocols specify the data rate and speed of operation. Thus there must be some way to handle the exchange of data between two devices and/or transmission lines having different clock speeds, data rates without inducing any errors.

The present work encompasses designing a high speed link buffer in application to HDLC processor based on PCI bus, for an E1-T1 link. In this the data is written into the buffer at one clock speed and data width, whereas data is read out at a different clock speed and data width, to increase the throughput. To support our project application, appropriate Asynchronous multi-channel FIFO architecture is selected for designing a high speed link buffer. Proper designing of this buffer not only helps to increase interface data rate but also

aids in efficient management of data.

III. OVERVIEW OF MULTICHANNEL FIFO

In this information age, versatility and adaptability of a computer system is important so that it can interface with a host of other devices operating at various clock speeds. The interface under consideration here is the interface of E1-T1 communication link with the PCI based system. [1] E1-T1 links are serial communication links that use ISO standard HDLC protocol at Data-Link layer to communicate with system. HDLC Controllers are the devices which perform the functionality of the HDLC Protocol. Typically the rate at which data is received on an HDLC channel by an HDLC controller (from the link) is higher than the rate at which it can deliver data to the system.[6] Same is true for a transmit HDLC controller. Also data on a communication link is naturally bursty in nature making the exact amount of data unpredictable. In addition to this amount of data which a computer system can receive or send differs from amount of data travelling on the link. Hence the HDLC controller needs an arrangement to temporarily buffer the data. Asynchronous FIFOs are commonly used to transfer data between two clock domains.Figure3. Shows the basic block diagram of a high speed multichannel high speed buffer put into use in HDLC processor. As shown in block diagram the high speed multichannel buffer (FIFO) binds the link side and the system side together.[4] The diagram actually has two FIFO modules, one in transmit path and one in the receive path.On the receive path the data coming from E1 link side is reformatted in HDLC format by HDLC controller. This data is input to the receive FIFO from the receive HDLC controller. The FIFO buffers the data in such a way that it can be read out by DMA controller on the system side. The DMA interface with system side is based on PCI bus.[8] The segmentation is always supports the basic models to identify the HDLC controller in FIFO block supports the design structure in single model.

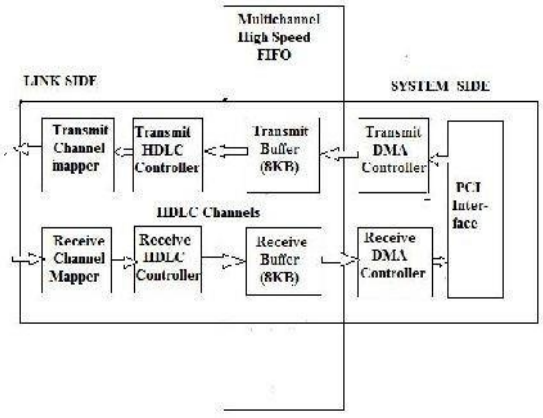


Fig3 Block Diagram of project using Multichannel FIFO

The system side and receive side operate at different data rates and clock speed. In the transmit path the transmit DMA controller prepares data to be written into the FIFO buffer. The FIFO buffers the data in such a way that it can be read out by HDLC controller on the link side.

IV. DATA STRUCTURE OF MULTICHANNEL-FIFO

The data structure of Multichannel FIFO is an important element and core of FIFO design. A multichannel FIFO has stack structure similar to that of RAM. It is logically made up of three main units namely data buffer RAM(DBRAM),block pointer RAM (BPRAM) and channel status RAM(CSRAM).These unit are interconnected through transport control and channel status controller as shown in figure 4.

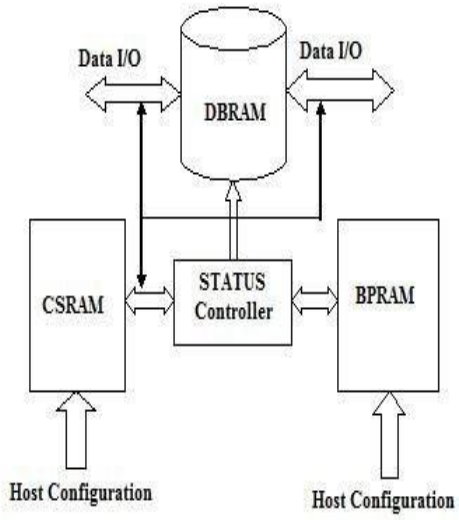


Fig 4. Logical Diagram of Multichannel FIFO

The DBRAM is organized as blocks of for efficient management of data. Each block is defined as 4 dwords that is 16 bytes. As a matter of fact DBRAM is composed of 512 such blocks. Each block is organized to form a FIFO, depth of which is 16 bytes and a width of 8 bits.

Used. Thus DBRAM is made up of 512 independent FIFOs. Associated with DBRAM is BPRAM which is organized as address space corresponding to 512 blocks of the DBRAM. Figure4.Describes the BPRAM's address space. Several blocks are link listed together to form a chain.[5] It is possible that all the channels are not of equal size and a varying number of blocks could be allocated to different channels. Maximum block size could be 512.In this case number of logical channel will be less than 128.Here we stick to the convention that all channels are allocated equal

number of blocks and each channel FIFO is made of 4 blocks.These channel FIFOs are assigned to take on the data by software configuration of the host system. In a nutshell the multichannel FIFO can be considered as lots of channel FIFOs, which take on uniform data width as well as different data depth and simultaneously perform data buffer management for different logical channels.

ADAPTED METHODOLOGY FOR DEVELOPMENT

Project Flow

The Top-Down design approach is used throughout the design of high speed link buffer .To implement any complex system, requires segmentation. The segmentation flow of this project shown is in project flow diagram 5.1 In order to reduce total time, the following processes are run in parallel:

The functionality of the design is verified with an advanced simulation tool Questasim and the results are saved for further authentication. To carry out the simulation a test bench module is developed. The test bench is like a wrapper code which accepts the inputs, feeds the inputs to different design modules which are instantiated and produces output waveforms. If the design and coding is done correctly the simulator will always produce expected output. The logic synthesis and formal verification of design is carried out in Xilinx ISE version 10.1 environments with the design code instantiated in same test bench used for simulation. This involves gate level synthesis, placing and routing and timing verification. Xilinx FPGA-Spartan X3S500e is used for successful hardware implementation and functional verification. Advanced visual tools are embedded in the ISE to analyse the output waveforms. Several logical channels are tested at key points in both transmit and receive directions. The simulation waveforms are compared to establish the success of the design.

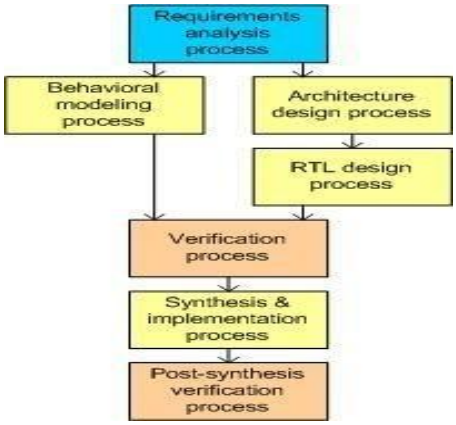


Fig. 5.1.Project Flow Chart

FIFO Transmit module:

In transmit direction FIFO module is separated into four sections; write state machine, read state machine, control state machine and channel poll state machine .After initializing the transmit FIFO write controller will continuously check whether DMA controller has any requests for free FIFO channels that control the identification of all three module for controlling the buffer system. The identification for changing the DMA and its identifier module controls the basic verification process in system control to change the sequence function and its sequence.

FIFO Receive module

The data coming from E1 link side is reformatted in HDLC format by HDLC controller. This data is input to the receive FIFO from the receive HDLC controller

.

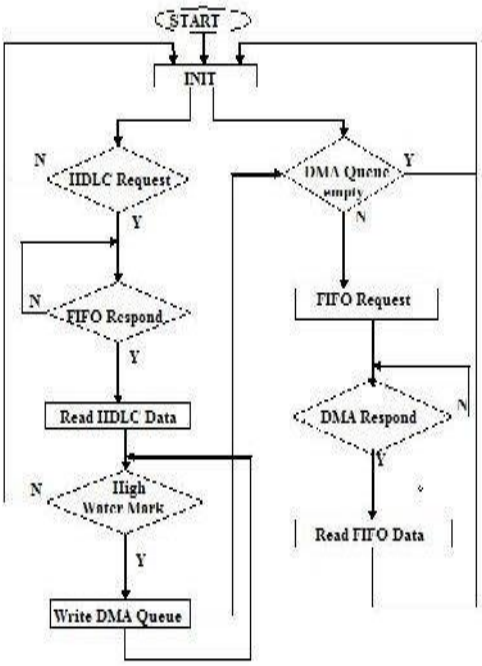


Fig5.2 Read/Write Sequence for receive FIFO module

The FIFO buffers the data in such a way that it can be read out by DMA controller on the system side. The DMA

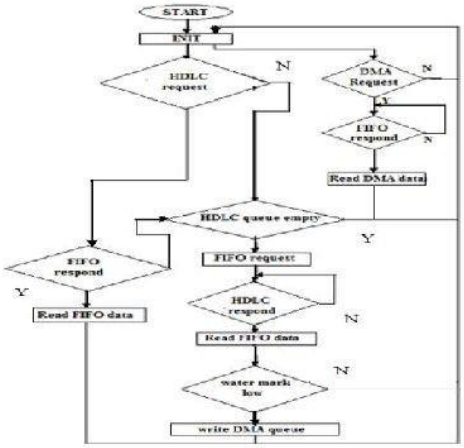


Fig 5.3. Read/Write Sequence for transmit FIFO module

If FIFO has enough memory free which can be assigned to DMA data, the write controller will start reading the DMA data fill up the FIFO. In the mean while FIFO read controller will allow the FIFO data to be read if there is a request from the HDLC controller. The data will now be sent on the link thereby making the FIFO empty.

V. IMPLEMENTATION

In Figure 6.1, RTL schematics for memory address decoding and a write operation..It has the necessary combinational logic for decoding a 5 bit memory address it also accepts an active high signal as a write enable in addition to address bits.

In Figure 6.2, RTL schematics for a receive FIFO Channel It gives overall view of a complete receive. FIFO including its write port, real port and necessary

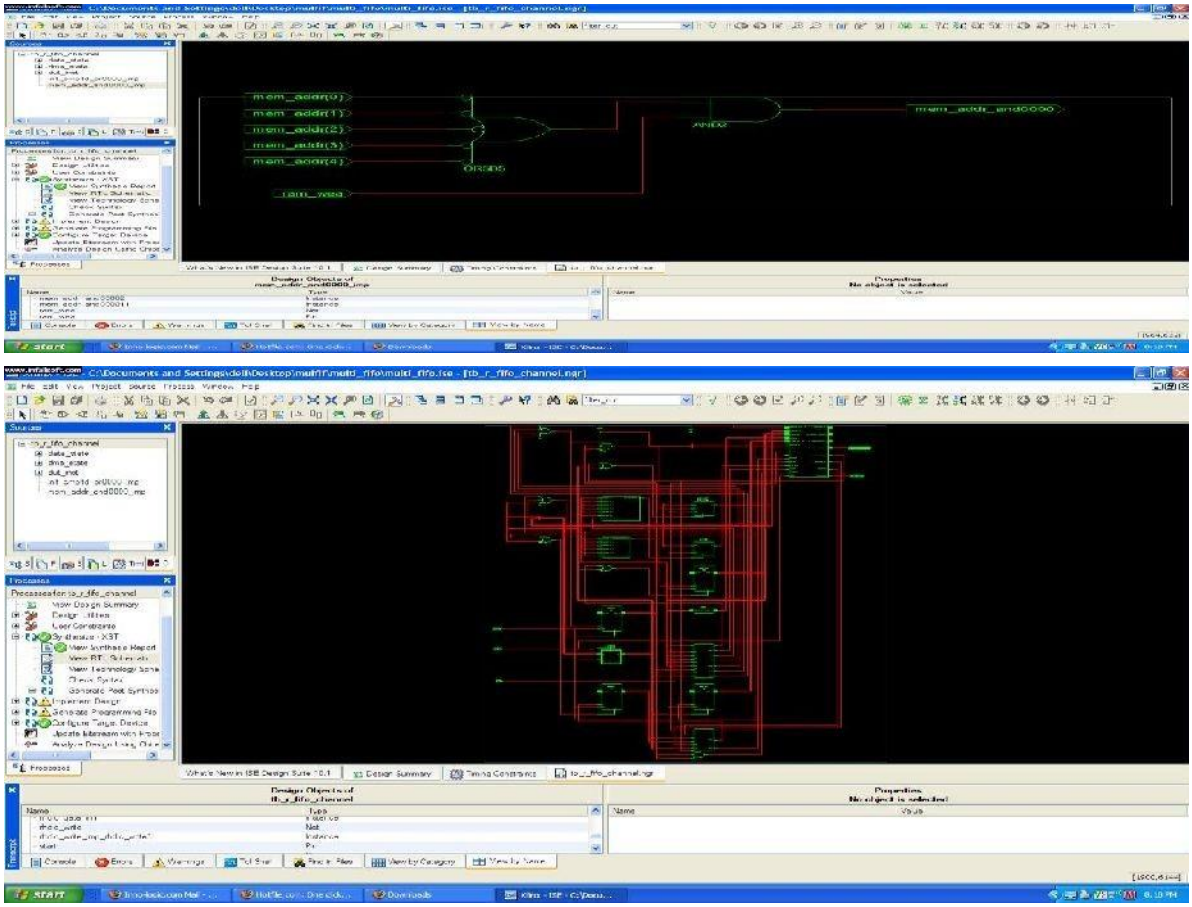


Figure 6.1. Shows the RTL schematics for a receive FIFO channel
Figure 6.2. the RTL schematics for a receive FIFO channel

VI. EXPERIMENTAL RESULTS

This chapter contains all the test results, on which the project statements are defined. Numerical values and graphical visibility are made use of for this purpose, which are obtained as shown below. Experimental results are grouped as simulation results and implementation results. First the simulation result are presented and analyzed in details.

This project is designed to implement 32 separate logical channels. Each channel is of 64 bytes

.Testing procedure is same for all the channels.

A. Simulation results for transmit and receive module

Figure7.1 shows the complete operation of a receive FIFO model. This includes initialization ,ead and write operation. Channel 0 is defined depth of which is 4 blocks and receive high water mark which is 2 blocks. As can be seen in figure, the initially high 'reset (r)' signal is made low between 175ns to 200ns which indicates that

initialization can begin.

Next between 525ns to 575ns HDLC processor sends request signal for Ch0 'channel_req' is set to 1.As soon as 'channel_req_ack' is received 'r_hdlc_write' is set and data is written into ch 0.

After this between 770ns to 780ns req_to_dma is set to 1, which indicates that the channel has reached high water mark and ch 0 makes a request to DMA controller. DMA sends acknowledge and sets dma_queue_rdy and dma_read indicating that DMA controller is ready to read from the FIFO. Between 940ns to 950 ns when rfifo_read_rdy is set to 1 it can be seen that data is being read from Ch1.Data is read continuously until block count is 0.

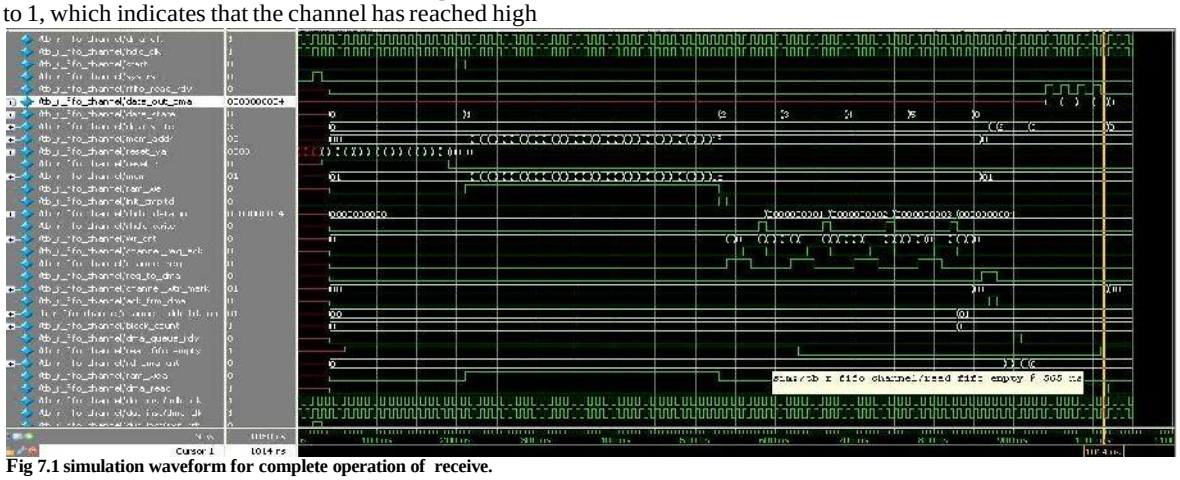


Fig 7.1 simulation waveform for complete operation of receive.

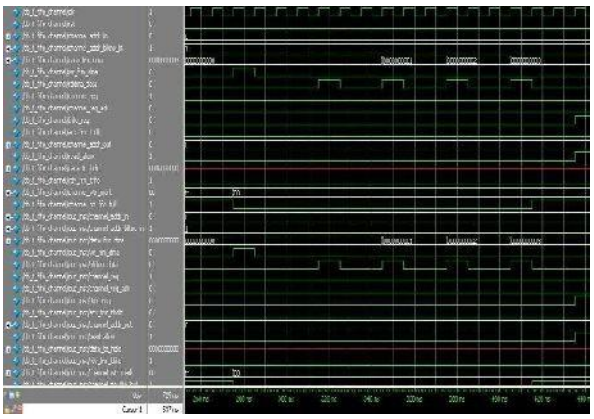
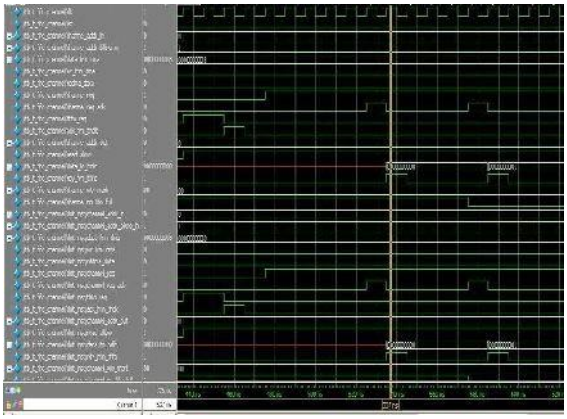


Fig 7.2 simulation waveform for transmit FIFO module for write operation.



Figs 7.3 simulation waveform for transmit FIFO read operation.

B. Implementation results

In this section output waveforms from the test board are presented. The code is synthesized

Figures 7.2 and 7.3 show the simulation results for transmit FIFO module. According to the sequence of events shown in fig7.3 the waveforms can be observed at different instances of time.

using Xilinx ISE synthesis tool to produce bit file of the code. With aid of Chipscope embedded within the ISE we can observe the output as shown in following figures Figure 7.4 show the output results of hardware implementation.

Data_port [0],data_port[1.data_port[39] are outputs which is of 40 bits. Here we can see all 40 bits which is present in left side of both the windows. In the centre of window all 40 bits are combined in one port by the name data_port which is your ultimate output port. Here you can see only 0000001 value because clock is running at 50 MHz. All read operation is done and only last value can be seen.

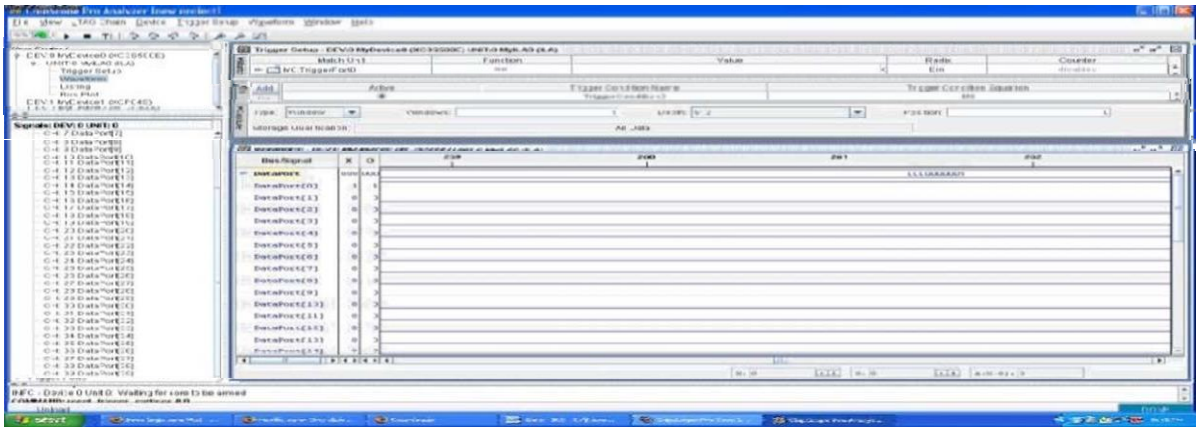


Fig 7.4. Chipscope output window2

VII. CONCLUSIONS

The multi-channel high speed FIFO is an important element and the core of the implementation of " Design of High Speed Link Buffer for HDLC processor based on PCI bus". This implementation gives the clear picture to implement a logical design of the high speed multi-channel FIFO on FPGA. Its functionality has been verified successfully with help of advanced synthesis tools .A general study of simulation waveforms at several important points proves that the proposed design gives a good performance and meets all the design requirements well within the tolerance. The multi- channel high speed FIFO designed in this project buffers HDLC data to DMA controller based on PCI bus. It is possible to write DMA wrappers such that the buffer implementation can be extended to support AMBA AHB and AXI bus as well as the device sizes shrink and power efficiency takes prime importance power consumption of the design has to be taken care of. It is possible to provide programmable FIFO channels which can reduce the power consumption.

REFERENCES:

[1] Bin Huang, Zhigong Wang, LuFeng Qiao, Yuanlin Lu, "Design of a High Multi-channelFIFO applied to HDLC processor",Institute of RF and OE ICs, southeast university, Nanjing, China ,IEEE 2002

[2] Clifford E. Cummings and Peter Alfke , "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons" Sunburst Design, Inc. SNUG-2002 San Jose, CA ,Xilinx, Inc.

[3] LogiCORE™ IP FIFO Generator v4.3User GuideUG175 March 24, 2008

[4] Tom Fischaber , "Never Design Another FIFO " Xcell Journal Third Quarter 2005 by Staff Design Engineer, IP Solutions Division Xilinx, Inc. Design Engineer, IP Solutions Division September 2005.

[5] Markus Levy , "FIFO memories supply the glue for high-speed systems ", Technical Editor EDN Access design feature, March 14, 1997.

[6] Technical paper on Clock domain Crossing;closing the loop on clock domain implementation problems, Candence. September 26 , 2008

[7] Hassan Kamgar, "Asynchronous FIFO controller",VLSI Tech INC,CA,USA May16, 2016

[8] w2.cadence.com/whitepapers/cdc_wp. For information on metastability , June

12th 2012.

[9] HDLC Controller Solutions with Spartan-II- FPGAs-Customer ntutorial.www.xilinx.com March 2000.