Design and Verification of Configurable Multi-channel DMA controller

Meet Dave[1], Santosh Jagtap[2]

PG Student [VLSI], Dept. of ECE, GTU PG School, Gandhinagar, Gujarat, India[1]

Design Engineer, WIPRO limited, Pune, Maharashtra, India[2]

ABSTRACT: This paper proposes a method of high-speed data transmission by AMBA Direct Memory Access controller using asynchronous first in first out (FIFO).Though interrupt-driven Input/Output devices have far better efficiency than programmed I/O, Processor still requires another intervention to transfer data from memory to peripheral and vice-versa to increase throughput. But both of these techniques suffer from **1.** Limited transfer rate **2.** Processor's requirement in executing instructions for each data transfer. Thus, for transferring a huge amount of data, a much more efficient method is needed. And that is **Direct Memory Access.** Generally it has four channels to transfer the data. So the data transfer is limited upto four devices. Hence we have come up with a configurable multi-channel DMAC so that any number of devices can be able to transfer the data by modifying the arbitration scheme. The transfer of data noticeably reduces the load on the processor. The design is implemented in Verilog Language. From the experiments it is clear that the customized DMAC design performs around 2~3 times faster than traditional DMA controller, and reduces the burden of the processor significantly.

KEYWORDS: Asynchronous FIFO, DMA, Channel Descriptor, Flexibility, Arbiter, Verilog, Simulation, AXI.

I. INTRODUCTION

DMA - Direct Memory Access is a technique that allows an I/O module to transmit and receive information right to or from main memory with bypassing the processor to speed up the memory operations. This complete series of action is handled by the chip known as a DMAC (Direct Memory Access controller).In the traditional processors, there were Four DMA channels and were numbered as 0, 1, 2 and 3. When the 16-bit ISA (industry standard architecture) expansion bus is introduced, Number of channels are increased. The ISA has been replaced by AGP and PCI expansion cards which are a lot faster.

Each of the channels needs two lines in order to function : One for the DMA controller which asks for access of buses from the processor and second line is for the processor to recognize that the DMAC is able to transmit data over the lines without disruption from the processor. For this purpose, DMA Controller must use the bus at the time when the processor doesn't require it or by cycle stealing.

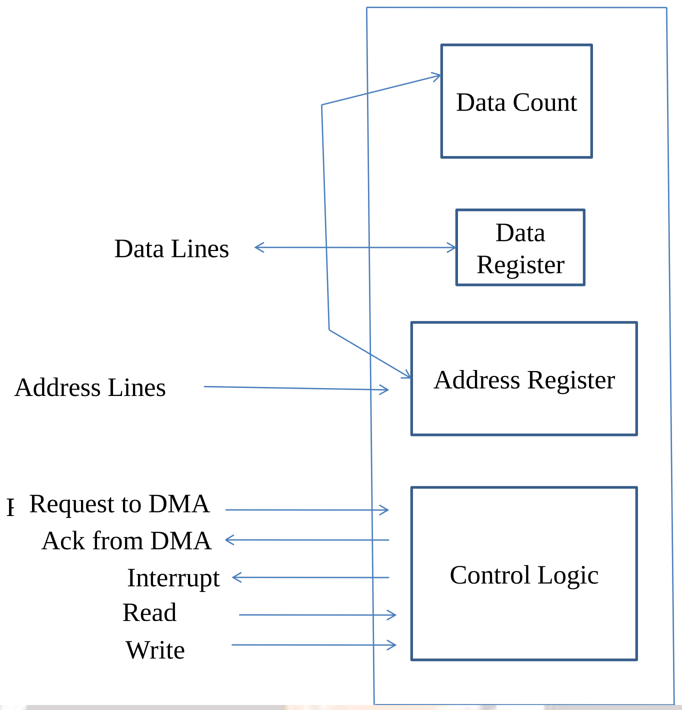

Figure 1 DMA block diagram

Figure 1 shows a typical block diagram of DMA. When a new device which wants to transfer large amount of data is connected to a processor, it issues few commands to a DMAC as shown in the figure. Then DMA takes the control over the buses in its hand and completes the data transfer to or from memory. After the transmission, DMA's control logic sends an interrupt to processor to take control of the buses.

In this paper, it proposes efficient communication between memory and processor with different arbitration scheme which provides more flexibility to the customized design. To overcome the shortcomings of traditional DMAC, an advanced DMA architecture with asynchronous FIFO is proposed for reducing the complexity and increasing flexibility with configuring more number of channels.

II.RELATED WORK

A. System Environment

From the processor point of view, the whole system is nothing but a memory. Hence for each operation that needs to be performed by processor, it requires the addresses of i/o devices, data width, starting location of the transmission of data etc. And all these information is stored in a Descriptor. Descriptor is nothing but a portion occupied in the memory module as shown in the figure 2.

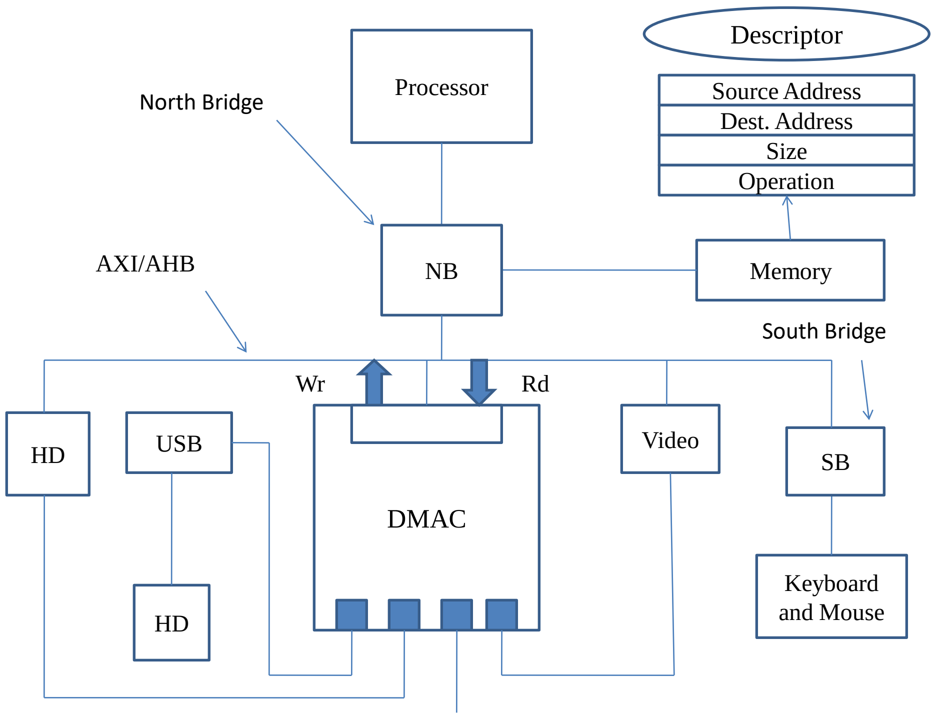

Figure 2 DMAC with actual Computer Architecture

When a device is connected to the system which wants to transfer data from or to memory, an interrupt occurs to the processor. So it stops its current task and provides the control of buses to DMAC after all required handshake signals shown in figure 1. Then DMA checks the descriptor for information related to different tasks on different DMA channels shown in figure 2. It is not mandatory that the information about source address, destination address, Size, Read/write operation etc are in sequence in memory for all four channels. There may be some other data in between. But it doesn't affect the DMA to achieve those information. Now with the help of arbitration scheme, DMA performs read/write transaction for different DMA channels.

B. Arbitration Scheme

We can come up with two Configurable arbitration Scheme :

1. Priority arbitration
2. Round Robin

It is a State Machine Kind of Mechanism.

0 - Highest Priority
1 - High Priority
2 - Low Priority
3 - Lowest Priority

III. SCOPE OF RESEARCH

Assume that you are downloading a movie from internet and you are getting speed of 2 MB/s. Now you are downloading a second movie at the same time. So what will happen? Speed will get divided by 2. The same thing happens when we are transferring data from DMA. The increase in number of devices, the decrease in speed. So we can improve the performance of DMAC by configuring it to eight channels from four channels and also by transferring data to AXI4 bus which is the latest peripheral bus providing few GB/s speed.

IV. PROPOSED ARCHITECTURE AND FUNCTIONAL OVERFLOW

Figure 3 shows the block diagram of DMA controller. Only few local bus interace is not shown here. Local bus can be AHB or PCI bus.

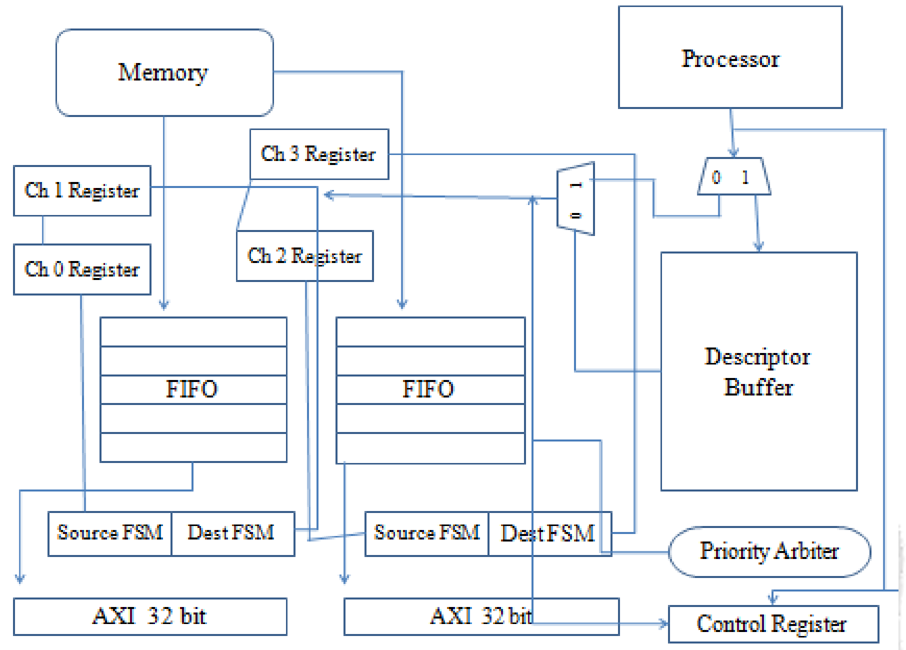

Figure 3 Block diagram of proposed DMA controller

DMA architecture is composed of 4 channels, descriptor, arbiter module, channel registers, three AXI masters, one AXI slave, address multiplexers and Asynchronous FIFO buffers, as is shown in Fig. 3. DMA controller is a four-channel structure which performs the specific data transmission. One AXI Master Interface is the only one master of its bus associated with the AXI protocol and two more AXI Master Interfaces can be connected to external memory or DDR. DMA controller works on two modes : 1. data transfer mode and 2. descriptor transfer mode. In the first one, DMAC transfers a bunch of data only and then selects a second priority channel. While in second one, processor sends tasks to channel descriptor ahead of schedule. After triggered, the highest priority channel is selected by arbitration scheme. By that, the first transfer information in the descriptor is loaded into the work registers and related transfer is executed. In parallel, the next data transfer information is programmed into the prefetch registers. This is how DMAC continuously transfers data without processor involvement til all programmed tasks in 4 channels are not finished.

V. EXPERIMENTAL RESULTS

In the experimental results, the graph of Asynchronous FIFO read and write transaction can be seen in figure 4. And those data are transmitted to DMA receiver and transmitter channel as seen in the figure 5 and 6.

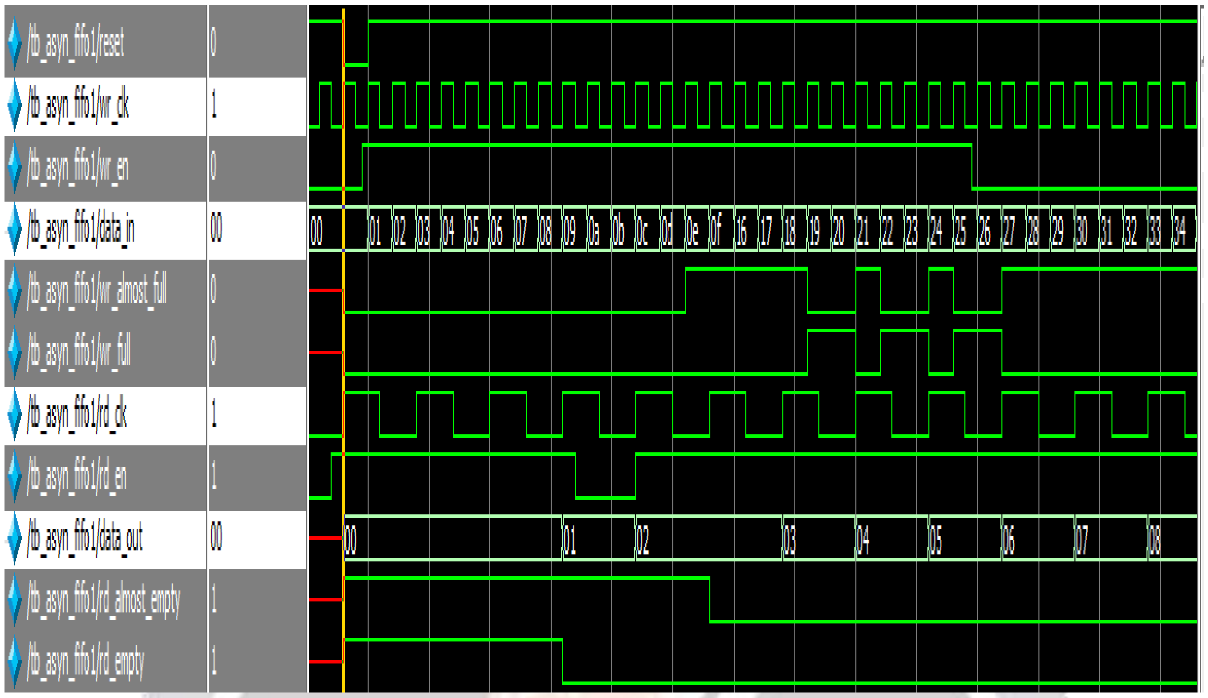

Figure 4 Simulation result of asynchronous FIFO

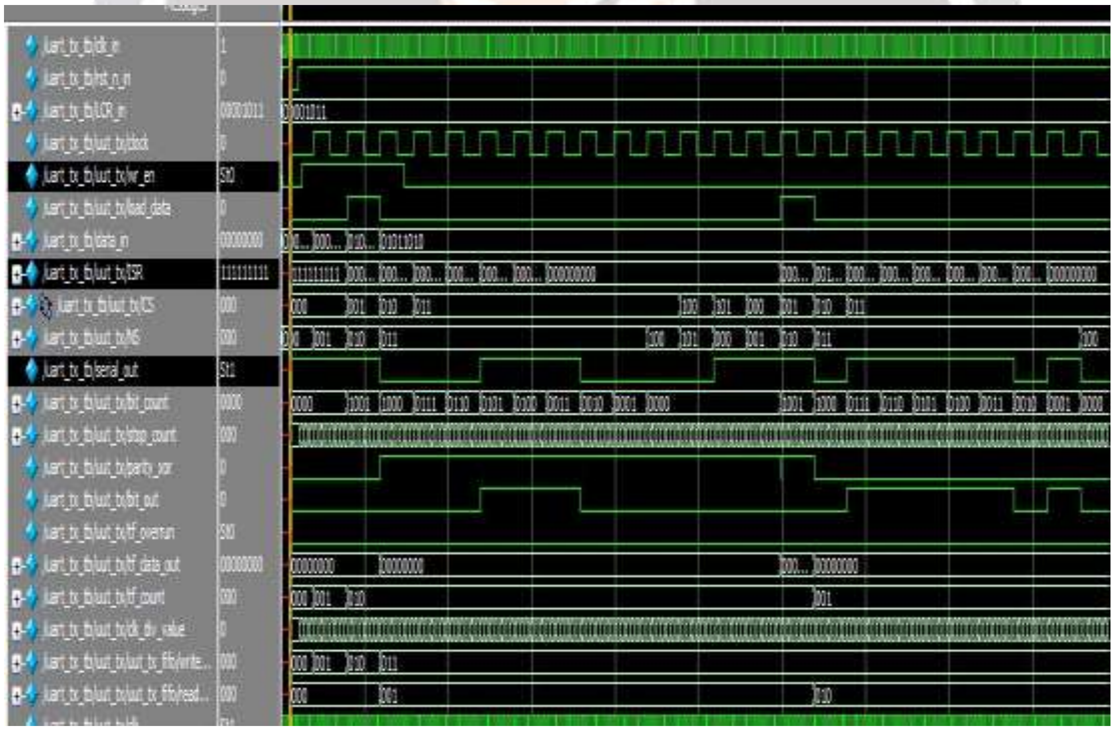
/uart_tx_tb/clk_in
/uart_tx_tb/rst_n_in
/uart_tx_tb/LCR_in
/uart_tx_tb/uart_tx/bclock
/uart_tx_tb/uart_tx/wr_en
/uart_tx_tb/uart_tx/load_data
/uart_tx_tb/data_in
/uart_tx_tb/uart_tx/TSR
/uart_tx_tb/uart_tx/CS
/uart_tx_tb/uart_tx/NS
/uart_tx_tb/serial_out
/uart_tx_tb/uart_tx/bit_count
/uart_tx_tb/uart_tx/stop_count
/uart_tx_tb/uart_tx/parity_xor
/uart_tx_tb/uart_tx/bit_out
/uart_tx_tb/uart_tx/tf_overrun
/uart_tx_tb/uart_tx/tf_data_out
/uart_tx_tb/uart_tx/tf_count
/uart_tx_tb/uart_tx/clk_div_value

Figure 5 Simulated Waveform for DMA Transmitter

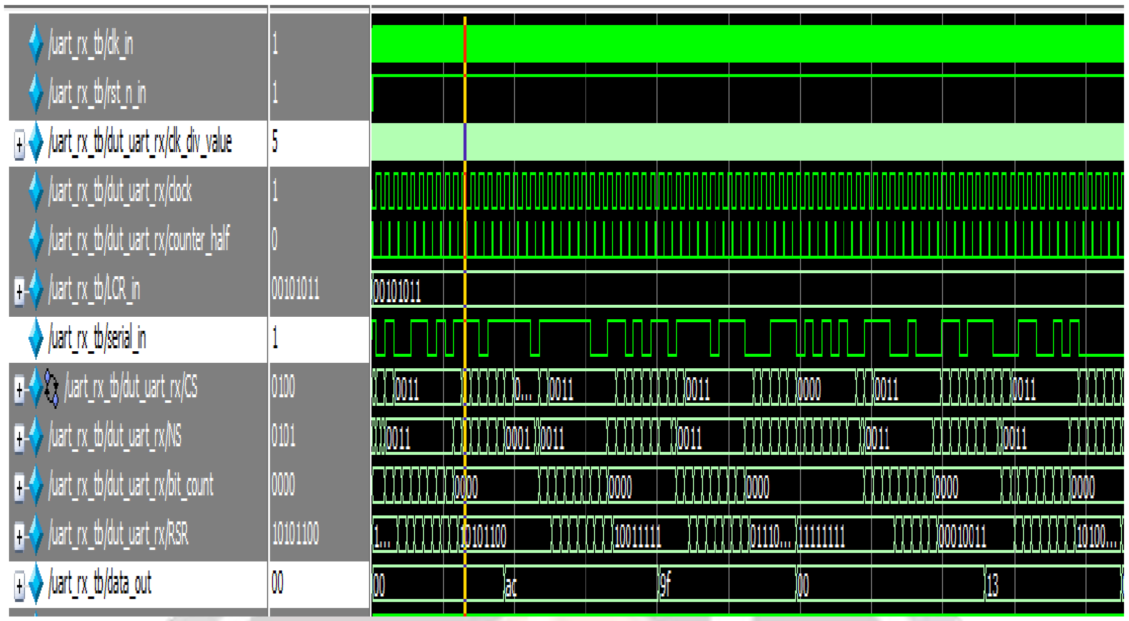
/uart_rx_tb/clk_in 1
/uart_rx_tb/rst_n_in 1
/uart_rx_tb/dut_uart_rx/clk_div_value 5
/uart_rx_tb/dut_uart_rx/clock 1
/uart_rx_tb/dut_uart_rx/counter_half 0
/uart_rx_tb/LCR_in 00101011
/uart_rx_tb/serial_in 1
/uart_rx_tb/dut_uart_rx/CS 0100
/uart_rx_tb/dut_uart_rx/NS 0101
/uart_rx_tb/dut_uart_rx/bit_count 0000
/uart_rx_tb/dut_uart_rx/RSR 10101100
/uart_rx_tb/data_out 00
00101011
0011
0000
10101100
10011111
01110...
11111111
00010011
10100...
ac
9f
00
13

Figure 6 Simulated Waveform for DMA reciever

.

VI. CONCLUSION

In this research paper, with the help of Priority arbitration scheme and asynchronous FIFO design Multi-channel DMA controller is simulated using Questa Sim 10.0b. With the new customized DMA controller, it will completely be flexible to configure a four channel DMA controller to eight, sixteen and so on easily depending on the system requirement. Throughput tremendously increases because of the configuration scheme as well as by transferring the data using AXI4 protocol.

VII. References

- PrimeCell Single Master DMA Controller (PL081) Revision Technical Reference Manual, AN2548 Application note.
- Liu Haihua, Chen Xinhao. "*Weighted Priority Rotational Algorithm on PCI Bus Arbitration*", Computer Engineering and Applications 2003.36
- H. Zhao, H. Sang, T. Zhang, X. Shen. “Design and Implementation of a Scalable Enhanced High Performance DMA Architecture for Complex SoC”, MIPPR: Medical Imaging, Parallel Processing of Images, and Optimization Techniques, Proc. SPIE, vol.7497, Oct, 2009.
- Y. Pan, J. Kim and G. Memik, "*Flexishare: Channel Sharing For An Energy-Efficient Nanophotonic Crossbar*," HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture, Bangalore, 2010, pp. 1-12.
- Vibhu Chinmay, Shubham Sachdeva (2014), “*A Review Paper on Design of DMA Controller Using VHDL”*, IJIRT, volume 1, issue6, ISSN: 2349-6002.
- Math, S.S., Manjula, R.B., Manvi, S.S. and Kaunds, P. “Data Transactions on system-on-chip bus using AXI4 protocol”, IEEE conference on Recent Advancements in Electrical, Electronics and Control Engineering (RAEeCE), December 2011, pp. 423-427
- Abdullah Aljumah, Mohammed Altaf Ahmed (2015), “*Design Of High Speed Data Transfer Direct Memory Access Controller for System on Chip Based Embedded Product's*, Ansinet journal of applied sciences, volume 15(3), pp 576- 581, 2015.
- Samir Palnitkar, Verilog HDL- A Guide to Digital Design and Synthesis, SunSoft Press, 1996, pp. 280-305
- Xinrui Zhang, Jian Wang, Yuan Wang, Dan Chen, Jinmei Lai “BRAM- based Asynchronous FIFO in FPGA with Optimized Cycle Latency ”, Solid-State and Integrated Circuit Technology (ICSICT), November 2012, pp. 1-3

- M. Sinnathamby and N. Manjikian, “A versatile Memory-Interface Architecture for Enhancing Performance of Video Applications”, International New Circuits and Systems Conference, IEEE, 2005, pp.91- 94.
- K. J. Lin, C. H. Huang, C. C. Lo. “Design and Implementation of a Schedulable DMAC on an AMBA-Based SOPC Platform”, Asia Pacific Conference on Circuits and Systems, IEEE, 2006, pp.279-282.
- N. Nandan. “High Performance DMA Controller for Ultra HDTV Video Codecs”, International Conference on Consumer Electronics, IEEE,2014, pp. 65-66.
- D. Katz, R. Gentile. “Using Direct Memory Access effectively in mediabased embedded applications”, Analog Devices, Inc. January, 2007