ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Design of High Speed DMA Controller using VHDL

Dharmik S. Dhamecha[1], Prof. Prashant R. Indurkar[2], Prof. Ravindra D. Kadam[3]

M. Tech (VLSI), Department of EXTC Engineering, BDCOE, Wardha, India[1]

Associate Professor, Department of EXTC Engineering, BDCOE, Wardha, India[2]

Assistant Professor, Department of EXTC Engineering, BDCOE, Wardha, India[3]

ABSTRACT: To meet rigorous high sustained bandwidth demand and exploit the data level access parallelism a new architecture of DMA controller for high speed data transfer is proposed in this paper. Trough analyzing the characteristic of data transfer; we customize four channel DMA with higher data carrying capacity. This customized DMA works in two different modes. In the first mode mode0 (Descriptor transfer mode) we can give a task table to descriptor buffer this consist of number of source and destination addresses for data transfer. In second mode mode1 (Data transfer mode) actual data transfer is done from source address to destination address trough the help of channels of DMA controller. Channels are selected as per the priority assigned to it. The proposed design code is written in VHDL language and it is synthesized and analyzed using XILINX 13.1 software. Experimental result shows that the customized design runes at clock frequency of 312.580 MHz and lower ratio of setting time and transfer time proves that the burden of processor is reduced significantly.

KEYWORDS: DMA, descriptor, arbiter, FIFO, channels.

I. INTRODUCTION

Direct Memory Access (DMA) is one of the several methods for coordinating the timing of data transfer between an input/output (I/O) device and CPU or memory in computer. An I/O device often operates at much lower speed than the CPU. DMA allows I/O devices to access memory directly, without using the CPU. DMA can lead to significant improvement in the performance because the data movement is one of the most common operation performed in processing applications.

In a computer system, DMA is a feature due to which the input/output devices can access the RAM of the computer independently of CPU. Without DMA, it is not possible for CPU to perform other tasks at the same time when the data is being transferred. But with DMA, it is possible for CPU to operate on other tasks during transfer of data. The CPU initiate the transfer of data. During the transfer of data between the I/O device and DMA channel, the CPU is independent to operate on other tasks. After the completion of transfer of data, an interrupt request from the DMA controller is received by the CPU. DMA is extremely important in those embedded systems where high performance, speed and multitasking is required because DMA makes the CPU free so that it can perform some other tasks which do not require data bus. DMA is generally used in those systems where a large chunks of data is transferred. In order to increase the parallism of system operations, it is necessary to use DMA controller instead of processor to accomplish data movement task. If the tradition DMA controller is used, plenty of time must be spent on the setting of DMA channel information and control information, which aggravates the burden of CPU and congestion on bus.

To remedy the shortcomings of tradition DMA, an optimized DMA architecture is proposed to gain flexibility and reduce complexity. Especially, descriptor buffer can accept a schedule request in advance in such a manner that can predominantly reduce the configuration times. The rest of the paper is organized as follows: section II describes the proposed DMA architecture and working followed design in section III. Section IV shows pin diagram and pin description. Section V contains the simulation result and section VI draws the conclusion.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

II. ARCHITECTURE AND FUNCTION OVERVIEW

The DMA controller is the coprocessor used for data handling, that is, it's controlled by CPU and acts accordingly. Whenever any application running on CPU need specific data transfer this will command DMA to do it. The address for data transfer is already decided by application and memory management unit running on CPU.

We hereby customize the DMA architecture composed of four channels, descriptor buffer, arbiter module, channel register and FIFO buffers as shown in fig 1. DMA controller is four channel structure which completes the specific data transfer.

DMA controller has two modes of operation, descriptor transfer mode (mode0) and data transfer mode (mode1). In the first mode (mode0) CPU sends the task table to the descriptor buffer ahead of schedule. The task table consists of number of source and destination address for required data transfer process. After that actual data transfer is enabled in mode1. The highest priority channel is selected by arbiter. Then the first transfer information in descriptor buffer is loaded into register of highest priority channel and corresponding task is executed. Then second transfer information is loaded in register of channel having lower priority. In this way, DMA controller keeps transferring data in conformity of task table without involvement of CPU, until all assignment in all channels are finished, marked by descriptor number 0. To adapt high speed transmission of real time or bursty data, asynchronous FIFO buffers are assigned to each channels of DMA controller.

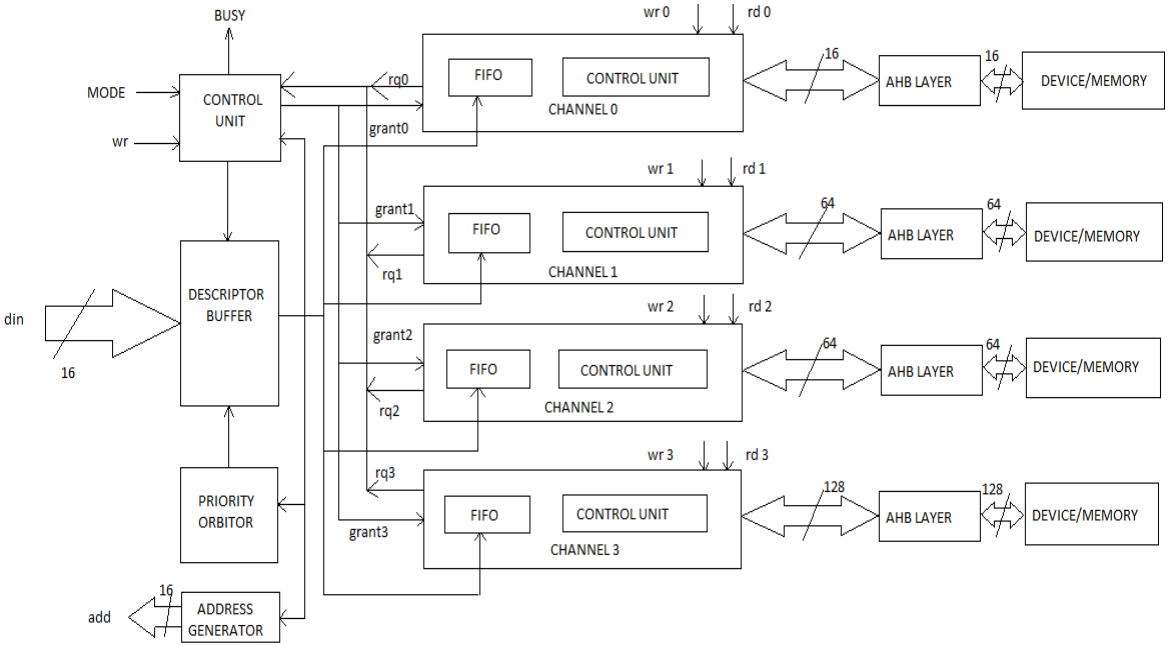

Fig. 2.1 Block diagram of DMA controller

III. DESIGN

I. CHANNELS OF DMA AND FIFO

The DMA controller has four DMA channels wiz ch0, ch1, ch2 and ch3. Each consists of two 16 bit registers for storing source and destination address for data transfer. It also consist of FIFO buffer to store the data to be transferred.

Before enabling a channel, the registers (both source and destination) must be initialized i.e. source and destination address from the task table which is stored in descriptor buffer must loaded into them. All the four channels are controlled by the control unit of DMA controller by request (rq0, rq1, rq2, rq3) and grant (grant0, grant1, grant2, grant3) signals. All the data transfer operation using channels are done in Data Transfer Mode (mode1).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

The sizes of four channel of DMA and there FIFO buffer is as shown in table 1.1

Sr. No.	Channel No.	Channel Size	FIFO Size
1.	Channel 0	16 bit	16x16 = 265 bit
2.	Channel 1	64 bit	64x16= 1024 bit
3.	Channel 2	64 bit	64x16= 1024 bit
4.	Channel 3	128 bit	128x16=2048 bit

Table 1.1 Sizes of channel and FIFO buffer

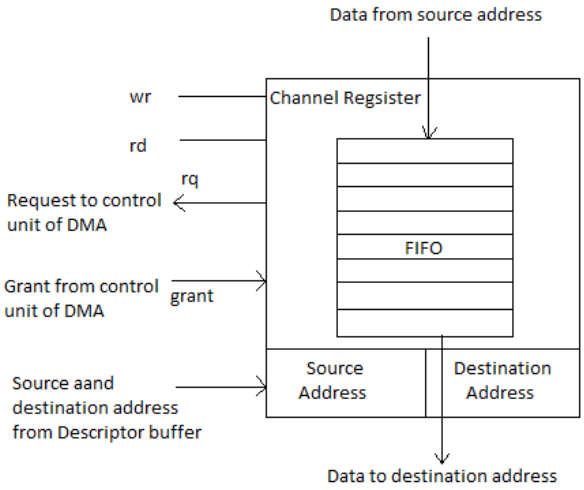

Fig. 3.1 DMA channel structure block diagram

As source and destination address is loaded into channel register, the channel will check the source address copy the data from source address and store in FIFO buffer. During this operation read pin of DMA controller for particular channel which is transferring data (rd0, rd1, rd2, rd3) is set. After that channel will check the destination address from the channel register and then store the data from FIFO buffer to destination address. During this operation write pin of the DMA controller for particular channel which is transferring data (wr0, wr1, wr2, wr3) is set. Source and destination address are auto incremented while data transfer. In this way all the four channels are used for data transfer.

II. ARBITER MODULE

DMA has four channels as mentioned above. We have assigned rotating priority to all four channels. The highest priority channel is channel 0 and lowest priority is assigned to channel 3.The flow of priority is as follows ch0, ch1, ch2 then ch3. Figure 1.3 shows the rotating priority of DMA channel.

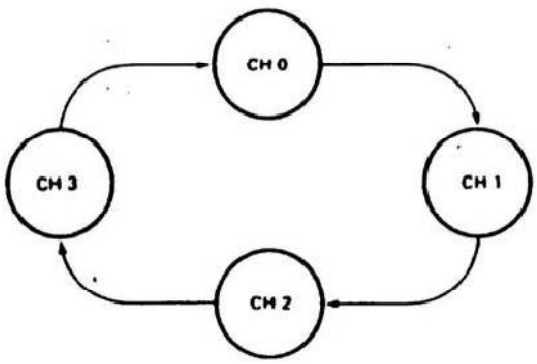

Fig. 3.2Rotating priority of DMA channels.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

III. DESCRIPTOR BUFFER

Descriptor buffer is two port structure, size of which is designed to contain up to 16 entries according to system data movement demand. Descriptor buffer is used to store the task table. Task table consist of set of source and destination address used for data transfer process. CPU can provide this task table ahead of schedule to descriptor buffer, which provides most flexibility in managing system transmission. DMA controller can set up its channels automatically and start another transfer after the current transfer is executed. Moreover, with the access address of each sequence calculated automatically, CPU can execute other control job on which it's supposed to be focused, largely reducing the burden of the processor.

IV. PIN DIADRAM AND PIN DESCRIPTION

Figure 1.4 shows the pin diagram of proposed DMA structure. Pin description is as follows:

1. Descriptor data in {din (15:0)}: This pin is used to store the task table in descriptor buffer in mode 0.
2. Channel data in pin {din0(15:0), din1(63:0), din2(63:0), din3(127:0)}: These are the channel data input pins, by using these pins data from source address is stored in FIFO buffer of particular channel which is currently enable for data transfer in mode1.
3. Clock: This is the clock signal input pin.
4. mode: Mode pin is used to set the modes of DMA. When mode = 0 DMA works in Descriptor transfer mode and when mode = 1 DMA works in Data transfer mode. Mode pin also acts as request from CPU to DMA for particular data transfer.
5. Descriptor write pin (wr): This pin is set before the descriptor buffer is loaded with task table from CPU.
6. Address generation pin {add(15:0)}: This pin is useful for address generation.
7. Channel data output pin{dout0(15:0), dout1(63:0), dout2(63:0), dout3(127:0)}: In mode1 these pins are used by channels to store the data from there FIFO buffer to destination address.
8. Busy: When DMA controller enters in mode 1 (data transfer mode) busy signal is set. This indicates that DMA is currently busy in transferring data. This pin also acts as acknowledgement to the CPU request form DMA for particular data transfer.
9. Channel read pin (rd0, rd1, rd2, rd3): This pin is set before data in read by particular channel from source address.
10. Channel write pin (wr0, wr1, wr2, wr3): This pin is set to indicate that channels are writing data from there FIFO buffers to destination address.

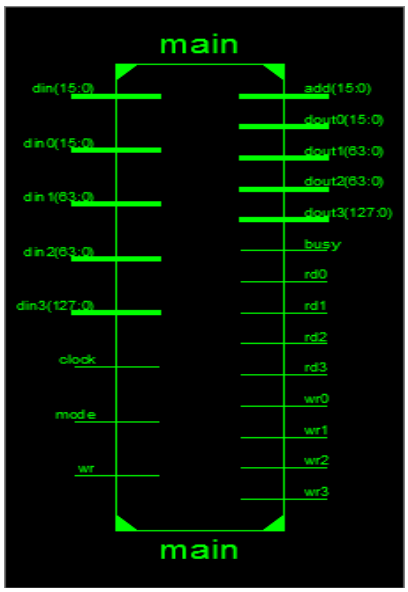

Fig. 4.1 Pin diagram of proposed DMA

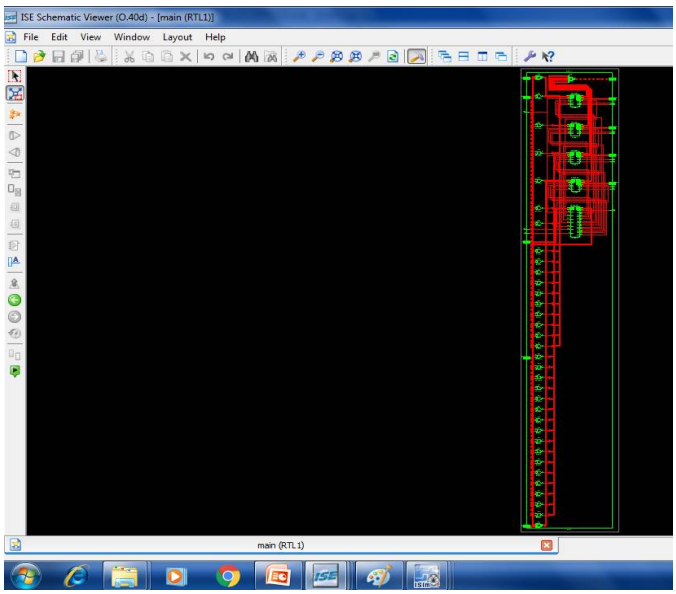

Fig. 4.2 Internal structure of proposed DMA.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

V. SIMULATION RESULT

In the simulation result diagrams shown below, simulation for all four channels simultaneously (fig. 5.1) and single channel (channel 0) (fig. 5.2) is shown. The system is synthesized for 3.199ns clock period, that is working frequency is 312.580MHz.

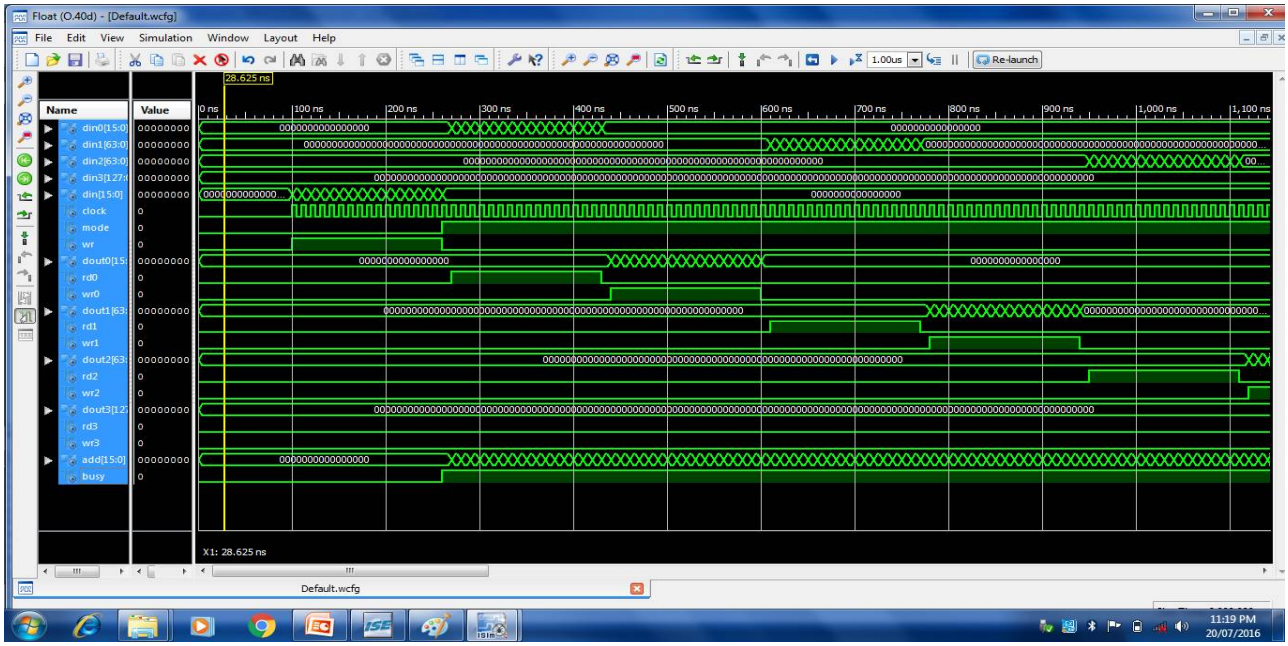

Fig 5.1Four channel simulation result.

While in simulation of any single channel, the delay is 1.612ns i.e. it can reach to maximum frequency of 620.347MHz.

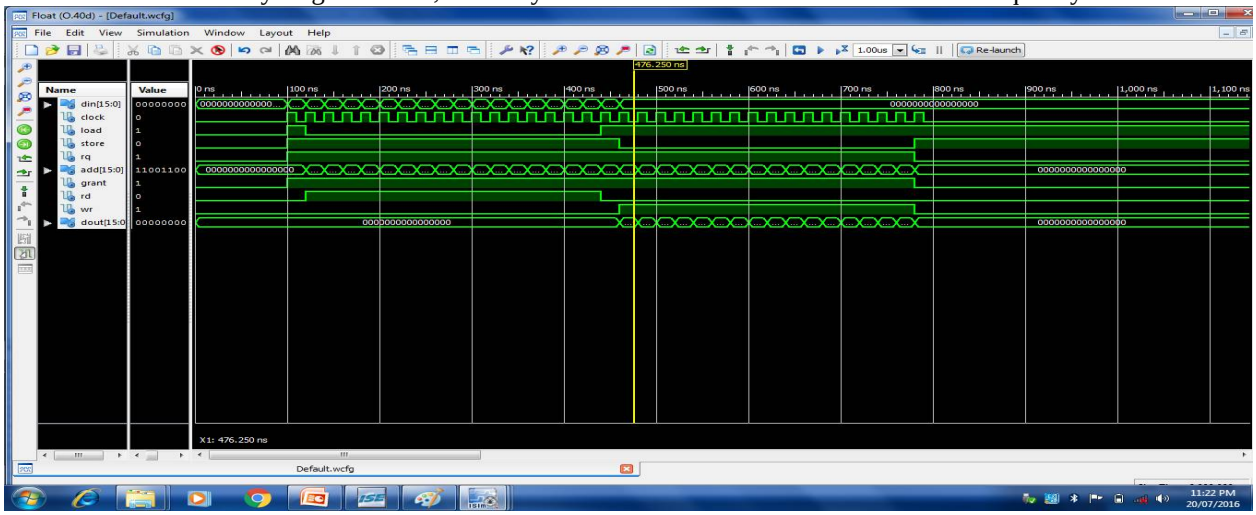

Fig 5.2 One channel simulation result.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

VI. COMPARISON

The proposed DMA controller is compared with other DMA structures on the basis of different parameters. Table 1.2 shows comparison.

Table 1.2

	1	2	3	4	5	6	My design
Design	Flexible DMA controller	Review on 8237 DMAcontroller	Hgh performance DMAcontroller	DMA memory copy accelerator	Smart DMA controller based on ARM1176 processor	Generic DMA	High speed DMA
Technology usde/ platform	Semic 65nm	VHDL/ XILINX 9.1	28 nm COMS	VHDL/ XILINX 9.1	Embedded Soc's	XILINX 9.2	VHDL/ XILINX 13.1
Frequency	250 MHz	12.5 MHz	266 MHz	----	100 MHz	50 MHz	312.580 MHz
Delay	4ns	80 ns	3.75 ns	----	10 ns	200 ns	3.19 ns
No. Of channels	4	1	4	----	----	4	4
Power	----	----	4mv	----	----	----	----
Area/ no. Of logic gates	23.6 k logic gates	----	0.5mm2	----	----	----	21.6 K logic gates
Application	Video codec system	Data transfer	Ultra HDTV video codec	Better speed performance	For LTE terminals	Data transfer	Data transfer

VII. CONCLUSION

VHDL language is very useful to design any kind of structure. To fulfill any systems demand of data transfer, this paper proposes a four channel DMA controller which is capable of many data transfer jobs. The use of four channel, descriptor buffer, arbiter module and FIFO buffers make the DMA controller more flexible and efficient for data transfer. Simulation result shows that proposed design using all four channels simultaneously works on frequency of 312.580MHz and can reaches up to maximum frequency of 620.347 MHz.Comparison result shows that the proposed DMA requires very less delay as compared to other DMA structures i.e. 3.19 ns. Hence the proposed DMA is high speed.

REFERENCES

[1] Yinhui Wang, Teng Wang, Pan Zhou, Xin'an Wang, "Design and Implementation of a Flexible DMA Controller in Video Codec System", IEEE / Proceedings of the 19thInternational Conference on Digital Signal Processing, 978-1-4799-4612-9/14/$31.00 © 2014 .
[2]Nilesh G. Bhoyar, YavandasDevgune, "Design DMA Controller Using VHDL", Discovery, Volume 18, Number 54, May 10, 2014.
[3]NirajNandan, "High Performance DMA Controller for Ultra HDTV Video Codecs", IEEE / 2014 IEEE International Conference onConsumerElectronics (ICCE), 978-1- 4799-1291-9/14/$31.00 ©2014.
[4]ArchanaTiwari&D.J.Dahigaonkar, "AMBA DEDICATED DMA CONTROLLER WITH MULTIPLE MASTERS USING VHDL", International Journal of Information Technology and Knowledge Management, January-June 2011, Volume 4, No. 1, pp. 285-288.
[5]Sebastian Hessel, David Szczesny, Felix Bruns, Attila Bilgic, Josef Hausner, ArchitecturalAnalysis of a "Smart DMA Controller for Protocol Stack Acceleration in LTE Terminals", 978-1-4244-3574-6/10/$25.00 ©2010 IEEE.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

[6]David Szczesny, Sebastian Hessel, ShadiTraboulsi, Attila Bilgic,"Optimizing the Processing Performanceof a Smart DMA Controller for LTE Terminals", The Sixteenth IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 1533-2306/10 $26.00 © 2010 IEEE.
[7]Anil Didariya*, Harshbardhan Singh Jasrotia, Rohan Gupta, SanjeevGurjar and Mr. NeerajTripathi,"Design and Implementation of Generic DMA using VHDL", International Journal of Current Engineering and Technology @2016, E-ISSN 2277-4106, P-ISSN 2347-5161.
[8] Direct Memory Access and DMA-controlled I/O
[9] Modern development of DMA

BIOGRAPHY

Mr. Dharmik S. Dhamecha achieved the B.E degree in Electronic and Telecommunication Engineering in the year 2014 from SGBAU, Amravati, India. He is currently doing MTECH degree in VLSI system design from BDCOE, Wardha, under RTMNU. His research focused on Design of high speed DMA using VHDL. He is winner of national level Tech-Fest organized by DESCOET college of Engineering. He has published one paper in research journal. His areas of interest are digital signal processing, Communication Engg. and VLSI.

Prof. Prashant R. Indurkar received the MTECH degree in Electronic from RTMNU, Nagpur, India. He is currently working as Associate Professor in the department of EXTC, B.D. College of engineering, Wardha, India and his total teaching experience is 24 year. He has published 40 papers in National / International conferences and journals to his credit. He is the member of the ISTE and IETE. His areas of interest are Digital, Embedded system and FPGA.

Prof. Ravindra D. Kadam received the Electronic in the year 1995 from RTMNU, Nagpur and the MTECH degree in Electronic C-DAC in the year 2011 from RTMNU, Nagpur, India. He is currently working as Assistant Professor in the department of EXTC, B.D. College of engineering, Wardha, India. His teaching experience is 12 year and also 2 year experience in industry. He has published 30 papers in National / International conferences and journals to his credit. He is the member of the ISTE and IE. His research focuses on the FPGA implementation of memory less V.D. with hybrid register exchange method. His areas of interest are signal system, VLSI, control system and FPGA.