Design of AHB Protocol Block for Advanced Microcontrollers

Rishabh Singh Kurmi
PG Scholar in TIT Bhopal, India.

Shruti Bhargava
EC Department TIT Bhopal, India.

Ajay Somkuwar
EC Department MANIT Bhopal, India.

ABSTRACT
The AMBA Advanced high performance bus (AHB) protocol design acts as an interface between two different IP cores. In this work initially the investigation on the AHB is carried out and the basic commands and its working are identified based on which the signal flow diagram and the specifications are developed for designing the AMBA-AHB using VHDL.
In this paper we propose the design and implementation of a flexible arbiter scheme for the AHB busmatrix based on burst operation. Basically, AHB burst operation is that a sequence of operation happens with respect to the size given and it supports only three burst sizes. The size is acting as one of the input to the master during the burst operation and after each burst operation, the master or slave will go to the IDLE stage.
The AHB design contains basic blocks such as master and slave and the working of these blocks based on arbitration scheme. According to arbitration scheme only one master can Access the bus at any one time. Multiplexer and Decoders are used to selects the appropriate signals between master and slaves that are involved in the transfer. This AMBA-AHB protocol can be adopted in all the applications provided the design should be an AHB compliant.

Keywords
AHB busmatrix, Arbiter, System on Chip, FSM for master and slave, Master and slave side arbitration, IP, VHDL.

1. INTRODUCTION
In recent days , the development of SOC chips and the reusable IP cores are given higher priority because of its less cost and reduction in the period of Time-to-market . So this enables the major and very sensitive issue such as interfacing of these IP cores. These interfaces play a vital role in SOC and should be taken care because of the communication between the IP cores property.

The on chip bus used in the SOC design by enabling the efficient integration of heterogeneous system components such as CPU_S , DSP_S , Application specific cores, Memories and custom logic. The SOC design requires a system bus with high bandwidth to perform multiple operations in parallel. To solve the bandwidth problems, there have been several types of high performance on chip buses proposed , such as AHB busmatrix from ARM , the PLB crossbar switch from IBM and CONMAX from silicore. Among them, AHB busmatrix has been widely used in many SOC designs. This is because of the simplicity of the AMBA bus of ARM which attracts many IP designers, and the good architecture of the AMBA bus for applying embedded system with low power. The AHB busmatrix is an interconnection scheme based on the AMBA AHB protocol, which enables parallel access paths between multiple masters and slaves in a system. This is achieved by using a more complex interconnection matrix and gives the benefit of both increased overall bus bandwidth and a more flexible system structure.
This paper work is chosen because currently the issues are increased in the industries due to the lack of proper data transferring between the IP cores on the system on chip. The communication between the different IP cores should have a lossless data flow and should be flexible to the designer too. Hence to resolve this issue the main aim of this work is to design an AMBA (Advanced microcontroller bus architecture) - AHB (Advanced High performance bus) protocol with burst operation and verifying its functional behaviour with the help of its simulation results.

2. MICROCONTROLLER STRUCTURE BASED ON AMBA-AHB
This type of microcontroller structure consists High perform--ance ARM processor , High bandwidth on chip RAM, High bandwidth external memory ,Direct memory access (DMA) device, Bridge as a converter ,UART ,Timer, Keypad , PIO and other devices based on application as shown in fig 2.1. An AMBA based microcontroller typically consists a high performance system backbone bus (AMBA-AHB), able to sustain the external memory bandwidth, on which the above given devices reside. This bus provides a high bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high performance bus is a bridge to the lower bandwidth APB, where most of the peripheral devices in the system are located.
The key advantages of a typical AMBA System are listed as follows.

- High performance
- Pipelined operation
- Multiple bus masters
- Burst transfers
- Split transactions

AMBA-APB (Advanced Peripheral Bus) provides the basic peripheral macro cell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically

- Have interfaces which are memory-mapped register.
- Have no high-bandwidth interfaces.
- Are accessed under programmed control.

The external memory interface is application-specific and may only have a narrow data path, but may also support a test access mode which allows the internal AMBA-AHB and APB modules to be tested in isolation with system-independent test sets.

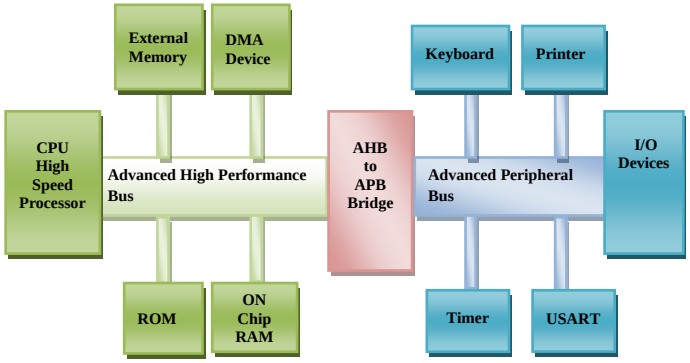

Figure 2.1: Typical AMBA System

2.1 AMBA-AHB Signals:-

All signals are prefixed with the letter H, ensuring that the AHB signals are differentiated from other similarly named signals in a system design. The signals involved in designing the AMBA AHB are listed below which also gives the specification of each signal.

- **HCLK:-** This signal contains width of 1-bit and it is driven by clock source. The data can be transfers at the rising edge of HCLK.
- **HADDR:-** Width of this signal is 32-bit and driven by master to assign address.
- **HTRANS:-** Width of this signal is 2-bit and driven by master. It indicates the type of the current transfer happening.
- **HWRITE:-** Width of this signal is 1-bit it is also driven by master. When high this signal indicates a write transfer and when low a read transfer.
- **HSIZE:-** Width of this signal is 3-bit and driven by master . It indicates the size of the transfer.
- **HBURST:-** Width of this signal is 3-bit, and driven by master. It indicates if the transfer forms part of a burst.
- **HWDATA**:- Width of this signal is 8-bit and driven by master. It is used to transfer data from the master to the bus slaves during write operations.
- **$HSEL_{X:-}$** Width of this signal is 1-bit and driven by decoder .Each AHB slave has its own slave select signal and this signal indicates that the current transfer is intended for the selected slave.
- **HRDATA:-** Width of this signal is 8-bit and driven by slave. It is used to transfer data from slaves to the bus master during read operations.
- **HREADY:-** Width of this signal is 1-bit and driven by slave. When high the HREADY signal indicates that transfer has finished on the bus. This signal may be driven low to extend a transfer.
- **HRESP:-** Width of this signal is 2-bit and driven by slave. It provides additional information on the status of a transfer.

3. ADVANCED HIGH PERFORMANCE BUS (AHB) PROTOCOL

The AHB is a high performance bus in AMBA (Advanced Microcontroller Bus Architecture) family. This AHB can be used in high clock frequency system modules. The AHB act as the high performance system backbone bus.

3.1 Enhancement of the AHB Design:-

- AHB is defined with a choice of several bus widths, from 8-bit to 1024-bit. The most common implementation has been 32-bit, but higher bandwidth requirements may be satisfied by using 64 or 128-bit buses.
- AHB used the HRESP signals driven by the slaves to indicate when an error has occurred.
- AHB also offers a large selection of verification IP from several different suppliers. The solutions offered support several different languages and run in a choice of environments.
- AHB can used be at higher frequency along with separate data buses that can be defined to 128-bit and above to achieve the bandwidth required for high-performance bus applications..
- AHB can access other protocols through the proper bridging converter. Hence it supports the bridge configuration for data transfer.
- AHB offers burst capability by defining incrementing bursts of specified length.
- It offers SEQ, NONSEQ, BUSY and IDLE transfer types.AHB also offers a fairly low cost (in area), low power (based on I/O) bus with a moderate amount of complexity and it can achieve higher frequencies when compared to others because this protocol separates the address and data phases.

3.2 Structure and operation of AMBA-AHB protocol

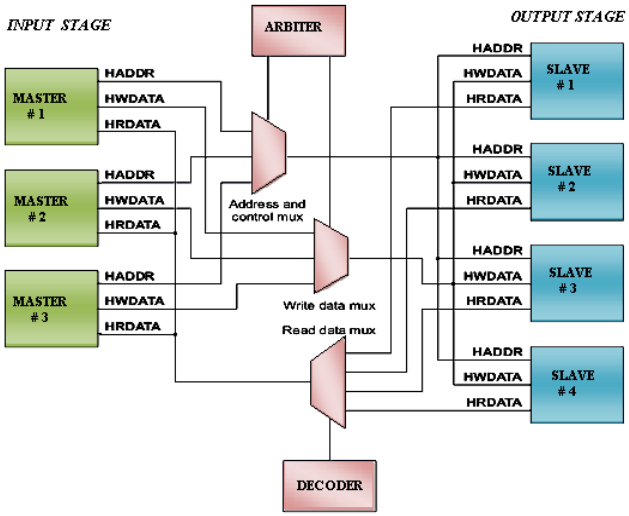

Figure.3.1: Structure of AMBA – AHB

The Advanced high performance bus AHB busmatrix of ARM consist of the input stage, decoder, multiplexor, arbiter and output stage. Fig. 3.1 shows the overall structure of the AHB busmatrix.

The input stage contains number of masters and output stage contains number of slaves to perform operation. The input stage is responsible for holding the address and control information.

Before an AMBA AHB transfer can commence, the bus master must be granted access to the bus. This process is started by the master asserting a request signal to the arbiter. Then the arbiter indicates when the master will be granted use of the bus. A granted bus master starts an AMBA AHB transfer by driving the address and control signals. These signals provide information on the address, direction and width of the transfer, as well as an indication if the transfer forms part of a burst. Two different forms of burst transfers are allowed.

- Incrementing bursts, which do not wrap at address boundaries
- Wrapping bursts, which wrap at particular address boundaries

A write data bus is used to move data from the master to a slave, while a read data bus is used to move data from a slave to the master.Every transfer consists of:

- An address and control cycle
- One or more cycles for the data.

The address cannot be extended and therefore all slaves must sample the address during this time. The data, however, can be extended using the HREADY signal. When LOW this signal causes wait states to be inserted into the transfer and allows extra time for the slave to provide or sample data.

During a transfer the slave shows the status using the response signals, HRESP **OKAY.** The OKAY response is used to indicate that the transfer is progressing normally and when HREADY goes HIGH this shows the transfer has completed successfully.

4. DESIGN OF AMBA-AHB WITH FSM

The literature survey on the AHB is made and the basic signal flow block diagram is identified. In the data flow signal diagram the basic signals are used in the simple read write and burst operation in AHB master and slave. Finite state machine (FSM) is developed and then modeled using VHDL.

The AHB takes on many characteristics of a standard plug-in bus. It's a multi-master with arbitration, putting the address on the bus, followed by the data. It has a data-valid signal (HREADY).

This bus differs in that it has separate read (HRDATA) and write (HWDATA) buses whose connections are multiplexed, rather than making use of a tri-state multiple connection. AHB can supports bursts with 4, 8, and 16 beat bursts and single transfers. But in this paper we are showing only 4-beat burst. The notations used while designing the AHB for the system Control signals are mentioned in the Table 4.1 and for others are listed in the Table 4.2 are as follows.

Transfer type (HTrans)

Table 4.1 Transfer type (HTrans)

HTrans	Notations Used	Description
00	IDE	No Data Transfer
10	NON_SEQ	The address and control signals are unrelated to the previous transfer
11	SEQ	The address is related to the previous transfer

HBurst values

Table 4.2 HBurst values

HBurst	Description
000	Represents Burst Size of 4

FSM for AHB master-burst operation

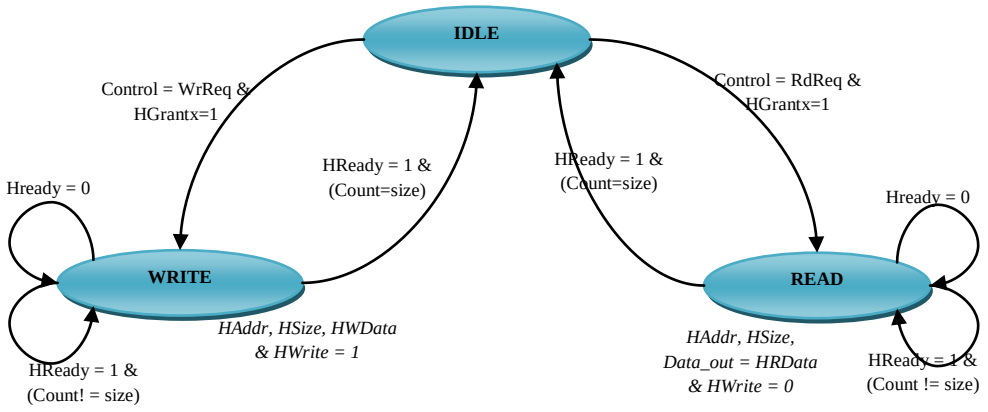

Figure 4.1 FSM for AHB master

FSM developed for the AHB Master – Burst operation is shown in the Figure 4.1 which has the clear view on the operation in a sequence manner. When write request (Control = "011") is given and the arbiter provides the grant signal in response to request signal, AHB Master goes to WRITE state. Here the count signal is added which will be incremented only when the HReady signal made high i.e. count increments after each operation and will remain in the same state. When burst size equal to the count, then the master goes to IDLE state that represents burst operation is over In Read operation, read request (Control = "100") is given which leads to the master to READ state in which the stored data in the memory is read out with respect to the given address to the slave. This output data sent through data out signal and when count is equal to the burst size, the master goes to IDLE state.

FSM for AHB slave-burst operation

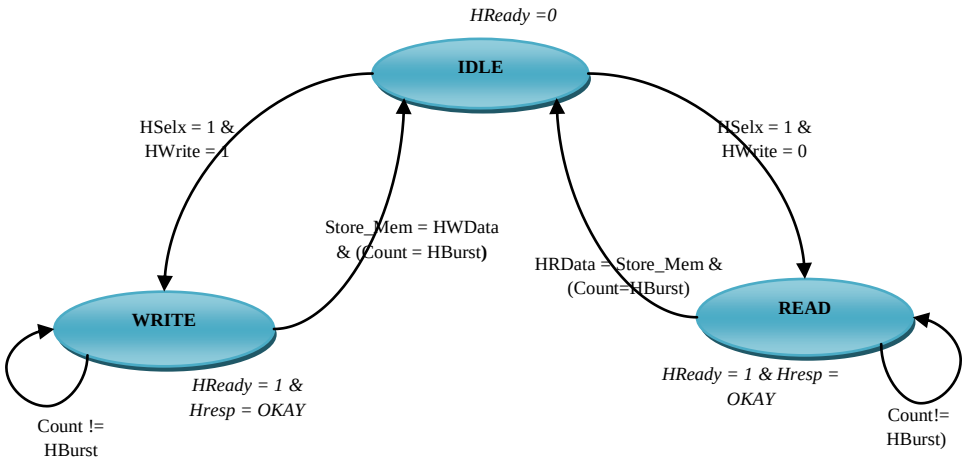

Figure 4.2 FSM for AHB slave

The FSM for the AHB Slave – Burst Operation is developed based on its operation and is shown in the Figure 4.2. In IDLE state HReady will be made low and when HWrite =1 slave will go WRITE state and check for the count. The count will get increment only when the data is stored in the memory and if count is equal to the burst size, then the slave will go to the IDLE state or it will stay in the WRITE state itself. In the same way, when HWrite = 0, slave will go READ state and the data is fetched from memory. Once the fetching process over, HReady is made high, HResp is set to 'OKAY' and count will check for burst size. Once the burst operation is over i.e. count is equal to burst size, then the count value resets to zero.

The figure 4.3 shows RTL Schematic of the proposed design is captured by Xilinx Synthesis Tool and it is representing the various signals used in the design.

4.1 Schematic of the AHB

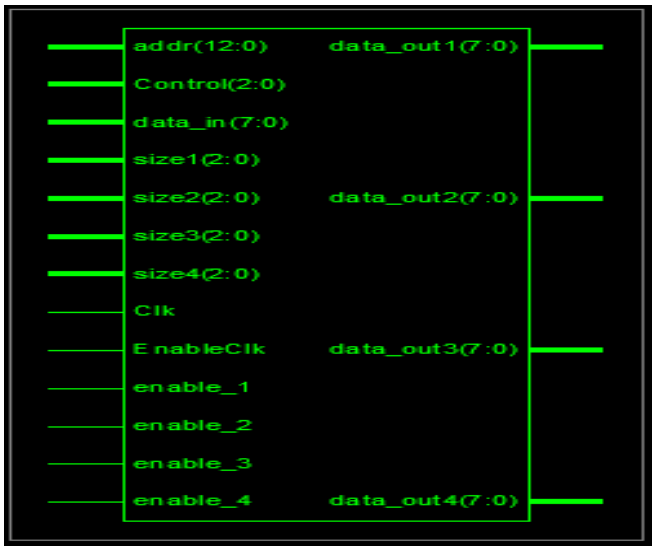

Figure 4.3 RTL Schematic view

5. SIMULATION RESULTS FOR BURST OPERATION

The basic working of AHB master and salve is discussed based on their FSMs and in the design totally four AHB master and slave are present. Here we are showing supports of AHB for burst size_ 4.

5.1 Burst operation of size_4

The simulation result for the AHB master and slave of burst size 4 is shown in the Figure 5.1.

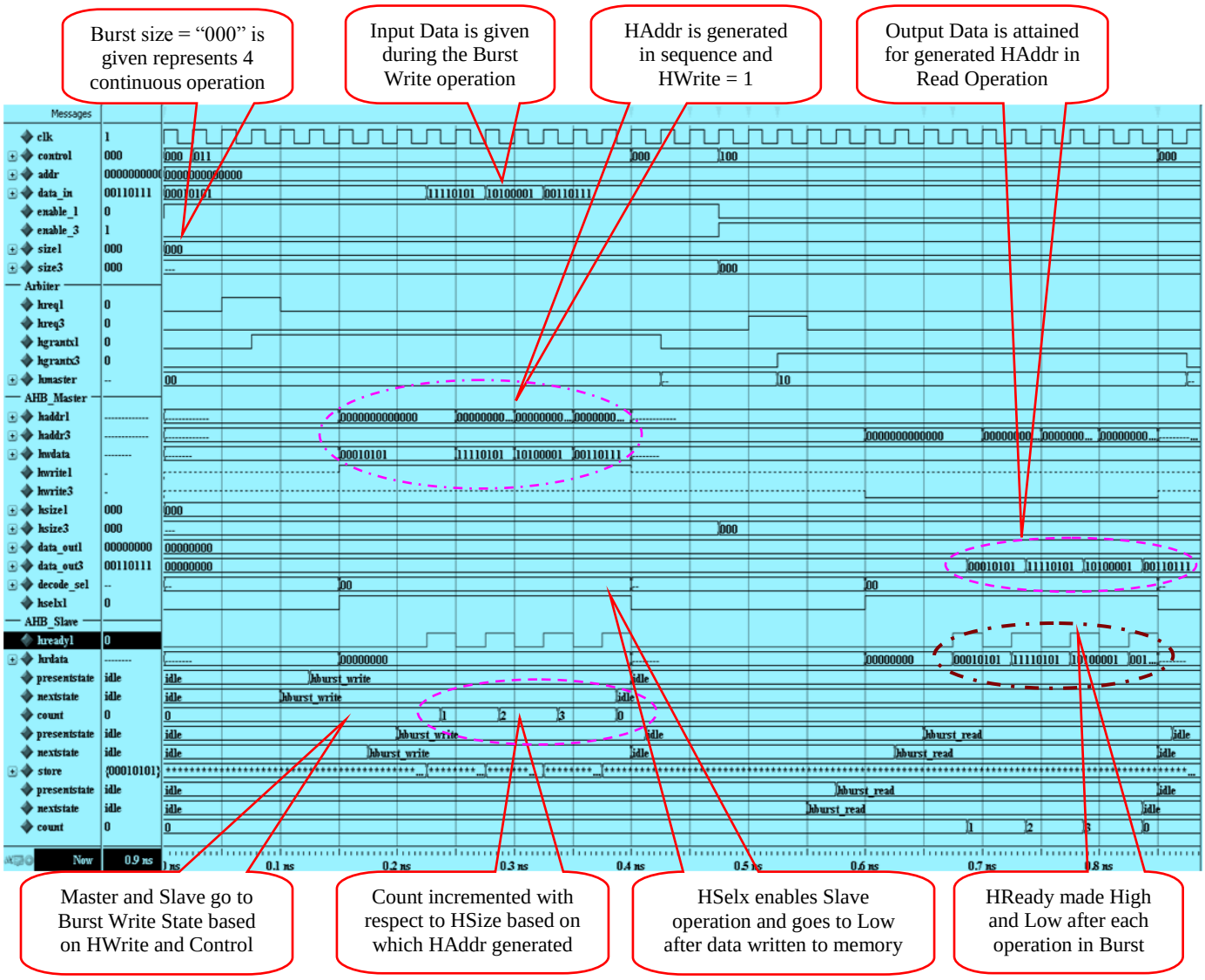

Figure 5.1 Waveform for AHB master and slave – burst operation of size 4

The size is given as “000” which represents the burst size 4 and hence four continuous write or read operation happens. Here the count is introduced in order to generate the address with respect the given initial address and the count increment. The operation remains the same as simple read and write but the only change is that after each operation, count will check for the burst size. When the count is not equal to the burst size given, the count will get incremented and the next address is get generated based on which the read or write operation that currently performed is carried out. When the count is equal to burst length, that represents the burst operation over and count resets to zero. Hence master and slave go IDLE state. HWright signal will be maintained as High or Low throughout the burst write or burst read operation and is made don’t care after the burst operation over. Similarly, the HReady signal is also made high for each operation in the burst which is clearly shown in the waveform.

The screen shots of the simulated waveform results shows that the communication between different IP cores using AHB is proper.

6. SYNTHESIS REPORT OF THE DESIGN

Final Results:-

Device Utilization	2s200eft256-6
Family	Spartan 2E
Device	XC2S200E
Package	FT256
Speed	-6

Design statistics	
IO_S	74

Cell Usage			
BUF	8	FD_1	32
GND	1	FDCE	18
INV	11	FDE	28
BELS	3911	FD	194
LUT1	248	FDPE	3
LUT2	344	FDRS	2
LUT3	472	FDS	121
LUT3_D	9	FDS_1	156
LUT4	1673	RAMS	48
LUT4_D	30	RAM32X1S	48
LUT4_L	130	Clk Buffers	1
MUXCY	364	BUFGP	1
MUXF5	324	IO Buffers	73
VCC	1	IBUF	41
XORCY	296	OBUF	32
Flip-flops	554		

Timing Summary Speed Grade :-6	
Minimum period	57.142ns
Maximum Frequency	17.500MHz
Minimum input arrival time before clock	17.792ns
Maximum output required time after clock	6.744ns
Maximum combinational path delay	28.571ns

7. CONCLUSION

- From the AMBA-AHB design we conclude that minimum period used in the design is 57.142ns and maximum frequency is 17.500MHz which is comparatively good result. The AHB is designed in such a way that the transaction between master and slave is carried out with proper delay and timings. We could design the intellectual properties of the master and slave depending upon specifications, data transfer and various transfer modes that are supported by AMBA bus architecture. Depending upon the real time application these intellectual properties can be used for designing high performance embedded microcontroller.
- The various scenarios for each component in the AMBA-AHB bus design are verified effectively during the simulation with respect to its behavior.
- We can use this protocol to interface between an ARM processor and any device (Like SRAM) provided both the IP cores should have AHB compliance. Data are successfully transferred from one IP core to other IP core i.e. no loss of data or control information.
- This paper work provides an ideal platform for the enhancement or further development of the AHB protocol.

8. REFERENCES

[1] Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang, Member, IEEE "Implementation of a self motivated arbitration scheme for the multilayer AHB busmatrix " VOL. 18, NO. 5, MAY 2010 IEEE .

[2] Rishabh Singh Kurmi, Shruti Bhargava " Implementation of an AMBA Advanced High performance Bus Protocol IP Block" IJECCE , June 2011, Vol.1, Issue 1, no. 2.

[3] S. Y. Hwang, K. S. Jhang, H. J. Park, Y. H. Bae, and H. J. Cho, "An ameliorated design method of ML-AHB busmatrix," ETRI J., vol. 28, no. 3, pp. 397–400, Jun. 2006.

[4] ARM, "AHB Example AMBA System," 2001 [Online]. Available: http:// www.arm.com/products/solutions

[5] IBM, New York, "32-bit Processor Local Bus Architecture Specification," 2001

[6] N.-J. Kim and H.-J. Lee, "Design of AMBA wrappers for multipleclock operations," in Proc. Int. Conf. ICCCAS, Jun. 2004, vol. 2, pp. 1438–1442.

[7] D. Flynn, "AMBA: Enabling reusable on-chip designs," IEEE Micro, vol. 17, no. 4, pp. 20–27, Jul./Aug. 1997 http://www.arm.com/products/solutions/axi_spec.html accessed Feb. 2008.

[8] S. Y. Hwang, H.-J. Park and K.-S. Jhang,"Performance analysis of slave-side arbitration schemes for the multi-layer AHB busmatrix," J.KISS, Comput. Syst. Theory, vol. 34, no. 5, pp. 257–266, Jun. 2007.

[9] C. H. Pyoun, C. H. Lin, H. S. Kim, and J. W. Chong, "The efficient bus arbitration scheme in SoC environment," in Proc. Int. Conf. SoC Real-Time Appl. Jul. 2003, pp. 311–315.

[10] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS on-chip communication architecture," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 6, pp. 596–608, Jun. 2006.

[11] S. Y. Hwang, H. J. Park, and K. S. Jhang, An Efficient Implementation Method of Arbiter for the ML-AHB Busmatrix. Berlin, Germany: Springer-Verlag, May 2007, vol. 4523, LNCS, pp. 229–240.

[12] Douglas l. perry, VHDL Programming.

[13] E Simon, An embedded design primer, pearson education.

[14] Rajkamal Microcontroller, ARM processor, pearson education. www.pearson.co.in.

9. AUTHORS BIOGRAPHIES:

Rishabh Singh kurmi, male , born in 2nd oct 1985 , home town Bina (M.P.) India , received his B. Tech in Electronics and Telecommunication Engineering from IETE ,New Delhi ,India in 2007, currently doing M. Tech degree in Microelectronics and

VLSI Design from TIT Bhopal, India. His areas of interest are Hardware Software Co-Design, Digital system Design and Embedded System Design, Analog Circuit Design.

Shruti Bhargava , female , born in 31st July 1985, home town Bhopal, India , received her B.E.Degree in Electronics & communication Engineering in 2007 from Globus Institute of Technology Bhopal, India, M.Tech in 2010 in Digital Communication from RKDF Bhopal, India.