DESIGN AND IMPLEMENTATION OF ADVANCED MICROCONTROLLER BUS ARCHITECTURE (AMBA) USING ADVANCED HIGH PERFORMANCE BUS (AHB)

Mrs.Dr.A.Mariajossy, CH.Manikanta, P.Nikhil, L,Mithilesh
Associate Professor
Department of Electronics and Communication Engineering
SRM Institue of Science and Technology
Kattankulathur, Tamil Nadu ,India

***ABSTRACT*—The Advanced Microcontroller Bus Architecture is widely-used for on-chip communication in System-on-Chip (SoC) designs. The AHB protocol provides a high-bandwidth, low- latency, and secure interconnect for connecting high- performance processors and high-speed peripherals within an SoC. The protocol has a standard with the interconnect topology, timing, and signaling, as well as support for multiple bus masters and slaves, burst transfers, and configurable arbitration schemes. The AHB protocol is widely used in applications such as microcontrollers, digital signal processors, and high- performance computing systems. Its robustness and versatility have made it a popular choice for complex SoC designs, and it continues to evolve to meet the needs of emerging applications. The circuit design is constructed and executed using Xillinx ISE and simulation analysis is done using ModelSim 6.4a.**

KEYWORDS - Advanced Microcontroller Bus Architecture; AHB;Arbiter;Decoder;Verilog;HDL; SOC;

I. INTRODUCTION:

In System-on-Chip (SoC) designs, the Advanced Microcontroller Bus Architecture (AMBA) is a popular on-chip communication protocol. The Advanced High-performance Bus (AHB) is one of several interconnect standards that make up the AMBA protocol family. One of the most popular protocols for trying together high-performance processors and high-speed peripherals inside a SoC is the AHB protocol.

The AMBA protocol family includes several connection protocols, one of which is the AHB. The AHB protocol is one of the most widely utilized for connecting high-performance CPUs and high-speed peripherals within a SoC.

AMBA protocols are open specifications. It is a need for interconnection. It is used in computer systems to connect and manage functional blocks. The AMBA controller optimises performance by converting the incoming signal to a memory controller protocol . It cuts down on the amount of pins.

Constructive system on chip must be assistive for manufactural testing and operation in order to limit the silicon infrastructure.

The research and aim of our project is to understand the design and implementing the architecture .

ABBREVIATIONS :

AHB - Advanced High Performance Bus

AMBA - Advanced Microconrtroller Bus Architecture

SOC - System On Chip

IP -Intellectual Property

II. LITERATURE SURVEY

The paper written by S Ramagundam et al. [1] appears to focus on the proposed structure and implementation of elements of the AMBA High-performance Bus (AHB) protocol, specifically with regards to the ACE (AXI Coherency Extensions) and slave interfaces with memory controller interface.
The paper describes the amba circuit and implementation of the AHB ACE and slave interfaces using VHDL, and provides detailed simulation results for area overhead and speed. The design is executed using the Xilinx 13.1 Spartan3 platform and is also simulated using ModelSim. The results of the study show that the AHB ACE interface has better performance than the AHB slave interface, with lower area overhead and higher clock speed.

The paper written by Yashdeep Godhal, Krishnendu Chatterjee, and Thomas A. Henzinger [2] presents a method for synthesizing AMBA AHB from formal specifications. The authors describe their framework, which includes an AHB Specification Language (ASL) for capturing formal specifications of AHB-based systems, a synthesis algorithm for automatically generating the corresponding AMBA AHB bus system, and a verification procedure for ensuring that the synthesized system is correct.

The paper presents experimental results of applying the framework to various case studies, showing that the synthesized AMBA AHB systems are functionally equivalent to the original AHB-based systems and are able to meet the required timing constraints.

This paper by Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang [3] presents an implemented an arbitration scheme that is determined for a multilayer AMBA AHB (Advanced Microcontroller Bus Architecture, Advanced High-performance Bus) bus matrix.

The authors describe the proposed scheme, which aims to improve the performance of the bus matrix by enabling the slave devices to prioritize their access requests based on their internal states. In the scheme, each slave device maintains a set of state variables that reflect its internal status, and uses these variables to calculate a priority value for its access request. The master device in the bus matrix uses these priority values to determine the order in which to grant access to the slave devices.

The paper by R. S. Kurmi and A. Somkuwar [4] presents the circuit architecture and implementation of an AMBA AHB IP to construct on chip controller.

The authors describe the architecture of the AHB protocol block, which consists of a number of sub-blocks such as the decoder, address decoder, data buffer, control logic, and error detection and correction circuitry. They also discuss the design choices and trade-offs involved in each sub-block, such as the choice of register-based or combinatorial logic, the size of the data buffer, and the complexity of the error detection and correction scheme.

The paper presents simulation results of the AHB protocol block, showing that it is able to achieve high throughput and low latency while maintaining correct operation under various test scenarios. The authors also compare their design with existing AHB implementations and show that it provides advantages in terms of simplicity, modularity, and flexibility.

The paper was written by M. Dubois and Y. Savaria [5] the paper discusses the design and implementation of a inclusive AMBA AHB for connecting high-speed design circuit in a system-on-chip (SoC) architecture. The authors describe the challenges involved in integrating multiple synchronous islands with different clock domains and present their solution based on the AMBA AHB bus standard. They also provide simulation results to demonstrate the performance of their design.

The paper was written by P. S. Shete and S. Oza [6] and In this paper, the authors propose a new Finite State Machine (FSM) design for implementing an AMBA AHB Master.

They highlight the limitations of existing FSM designs and propose a new design that overcomes these limitations. The proposed design is based on a new encoding scheme that reduces the number of states required for the FSM. The authors also present simulation results to demonstrate the performance of their design.

The paper was written by by D. Jayapraveen and T. G. Priya [7] and in this paper, the authors propose a design for a peripheral memory controller on the AMBA AHB protocol. The peripheral controller is designed to make a improvement in the overall system by efficiently handling the data transfer between the memory and other components of the system. The proposed circuit is executed by understanding and implementing VHDL and simulated using ModelSim. The authors present the simulation results to demonstrate the performance of their design in terms of delay, throughput, and power consumption.

DESIGN ARCHITECTURE OF AMBA BASED MICROCONTROLLER

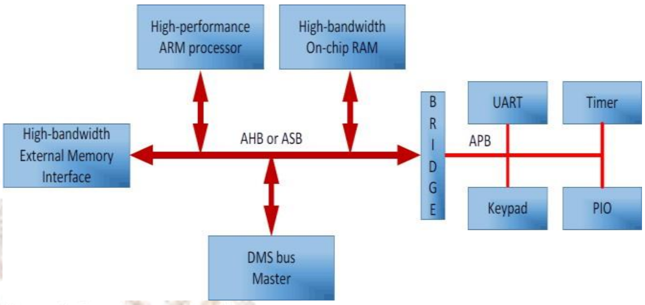

High-performance ARM processor
High-bandwidth On-chip RAM
High-bandwidth External Memory Interface
AHB or ASB
BRIDGE
UART
Timer
APB
Keypad
PIO
DMS bus Master

Fig 1 Design Architecture of AMBA Microcontroller

The standard design of AMBA AHB system has the showing components:

A. AHB MASTER: The AHB master can be a processor or any other device that is capable of initiating transactions. It has the ability to send write requests or read requests to a slave device, which could be a memory, a peripheral or another device.

B. AHB SLAVE: (AHB) Slave is a device that responds to transactions initiated by an AHB master. The AHB slave can be a peripheral device or any other device that is capable of receiving read or write requests.

C. AHB ARBITER: The AHB arbiter is responsible for resolving contention between multiple AHB masters that are requesting access to the bus. The arbiter prioritizes these requests and grants approach to bus to the highest priority in the design master.

D. AHB DECODER: AHB decoder (AHB) is a circuit that directs read/write requests master whichever proper needed slave device. It examines the address and control signals of the incoming transactions and uses this information to select the target slave device for the transaction.

E. AHB MULTIPLEXER: The AHB multiplexer is responsible for selecting one of several AHB bus masters to access the AHB slave. The AHB multiplexer acts as a switch, allowing one AHB master to communicate with an AHB slave at a time.

III. OPERATION OF AMBA AHB

The AMBA AHB protocol is a high-performance bus that works in conjunction with a central connectivity multiplexer system. AHB is made up of several masters and slaves that are linked together by an arbitrator and decoder.

Each exchange implies a control cycle and address and , as well has an one information cycle. Because address is impotent to be enlarged, in the midst of this period, all slaves the sample address must be used. The hready flag, on the other hand, can be used to extend the information. Whenever the flag is set to zero, the state of its to abide the incorporated into transfer of making the slave to provide the data sampled for a prolonged length of time. Using HRESP the signal which is used for response [1:0], the slave displays the data transfer and its status.
OKAY: The OKAY reply indicates that transfer is proceeding as standard data, and whenever the hready is

prominent enough that shows that the transfer itself has been done precisely.

This ERROR command reply indicates the interchange mishap happened, as well as that the interchange was failed. RETRY and SPLIT: The RETRY well as SPLIT exchange replies show the exchange could not be completed fast, though the master always makes tries to make the transfer. The HTRANS[1:0] signals reveal that

each transfer may be arranged into one of four unique categories.
The tabular indicates the types of transfers can be done.

HTRANS	*Type*	*description*
00	IDLE	Specifies no transmission is mandatory.
01	BUSY	Indicates that the bus master is continuing with the burst of transfer.
10	NONSEQ	Indicates the first transfer of a burst or single transfer.
11	SEQ	All the rest of the transfer is consecutive. The deliver will be identified with the past exchange.

TABLE 1 TYPES OF TRANSFERS

IV. INTERFACING OF AHB MASTER AND SLAVE

In the past years, a few techniques in the field of data sharing have been entirely established.

A. AHB MASTER: An AHB bus master provides the most sophisticated AMBA bus interface. The master uses the HBUSREQ signal to request the bus from the arbiter. Whenever the channel gets available arbitrator approves the request, and the grant signal increases, signalling that the bus reached the destination The master connects with the slave’s memory address by implementing the slave. Read and write operations are possible. "All operations are carried out on the positive clock edge."
Figure 2 depicts the AHB master interface design.

B. AHB SLAVE: An AHB slave communicates to the transfers that masters made inside the circuit design." The confirmation when to counter back data to a bus transfer, the slave utilises to pick the HSELx decoder that the signal needs to be used. All additional communication signals, including as address and control information, are generated by the bus master. The recognition signal is used to determine whether or not the communication was successful by sending a slave signal back to the master. Whenever a read operation is requested, data from the specified address is read and its sent back to the master. When writing is most desired and necessary, the memory place data must be identified and recognisable Every request is signalled to the master by the slave."
Figure 3 depicts the AHB slave interface design.

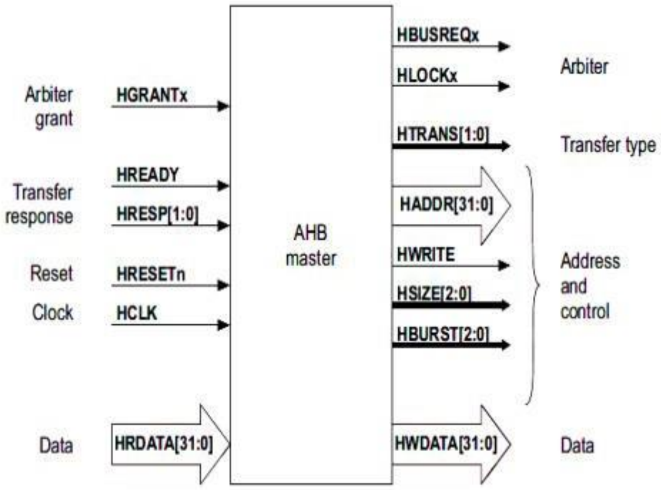


Figure 2 Design of AHB Master

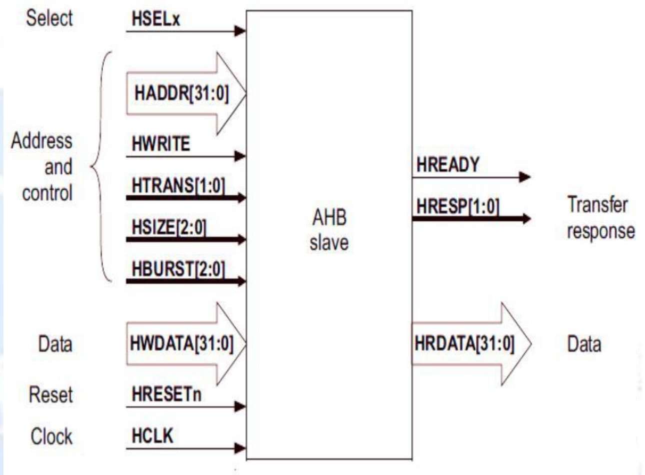


Figure 3 Design of AHB Slave

V. OTHER COMPONENTS OF AHB

C. AHB ARBITER: AHB Arbiter is implemented that it should pick a specific master from the several masters that are now seeking the bus. To resolve the conflict between the many asking masters, a fixed priority method is employed. In the following cycle, it additionally asserts the equivalent HMASTER signal for the given master, which is utilised by the mux to choose the signals of the selected mast. The arbiter uses the HSPLITx signal to restore a grant to a master that has previously been revoked owing to a split response by a split capable slave.

D.AHB DECODER: The AHB decoder typically receives the address signals and control signals from the AHB bus, and uses them to generate select signals to enable the appropriate slave devices. The decoder typically consists of a combinational logic circuit that generates a select signal for each slave device based on the address signals and control signals received.

E.AHB MULTIPLEXER: In AHB multiplexer are used for addressing and controlling , write data and read data multiplexers are implemented. The arbiter's HMASTER signal controls the address and control muxes, as well as the write data muxes, and the decoder's HSELx signal controls the read data muxes.

VI. SIMULATIONAL RESULTS

A. MASTER OPERATION: Fig 4 describes about the register transfer level of AHB master operation and Fig 5 describes the simulational waveform of master.

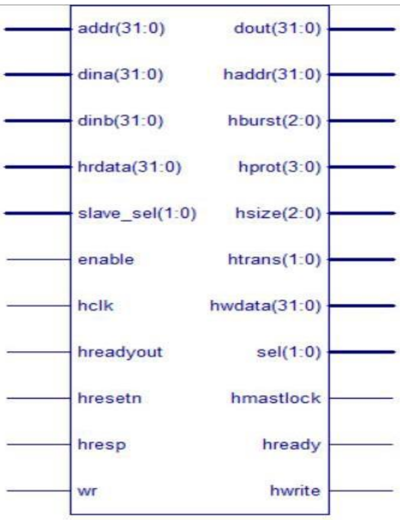
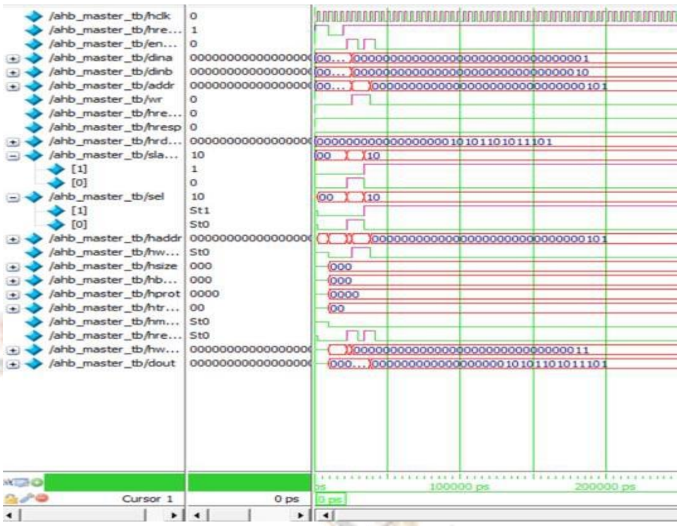


Figure 4 RTL Schematic of AHB Master



```
/ahb_master_tb/hclk      0
/ahb_master_tb/hre...    1
/ahb_master_tb/en...     0
/ahb_master_tb/dina      000000000000000000(
/ahb_master_tb/dinb      000000000000000000(
/ahb_master_tb/addr      000000000000000000(
/ahb_master_tb/wr        0
/ahb_master_tb/hre...    0
/ahb_master_tb/hresp     0
/ahb_master_tb/hrd...    000000000000000000(
/ahb_master_tb/sla...    10
  [1]                    1
  [0]                    0
/ahb_master_tb/sel       10
  [1]                    St1
  [0]                    St0
/ahb_master_tb/haddr     000000000000000000(
/ahb_master_tb/hw...     St0
/ahb_master_tb/hsize     000
/ahb_master_tb/hb...     000
/ahb_master_tb/hprot     0000
/ahb_master_tb/htr...    00
/ahb_master_tb/hm...     St0
/ahb_master_tb/hre...    St0
/ahb_master_tb/hw...     000000000000000000(
/ahb_master_tb/dout      000000000000000000(
Cursor 1                 0 ps
```

Figure 5 Simulational Wave of AHB Master

B. SLAVE OPERATION : Fig 6 describes about the register transfer level of AHB slave operation and Fig 7 describes the simulational waveform of slave.

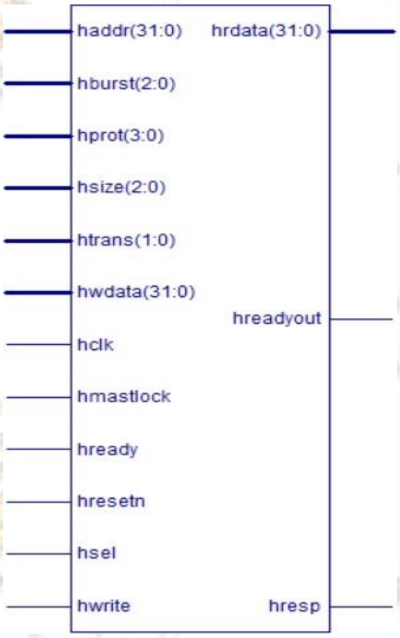
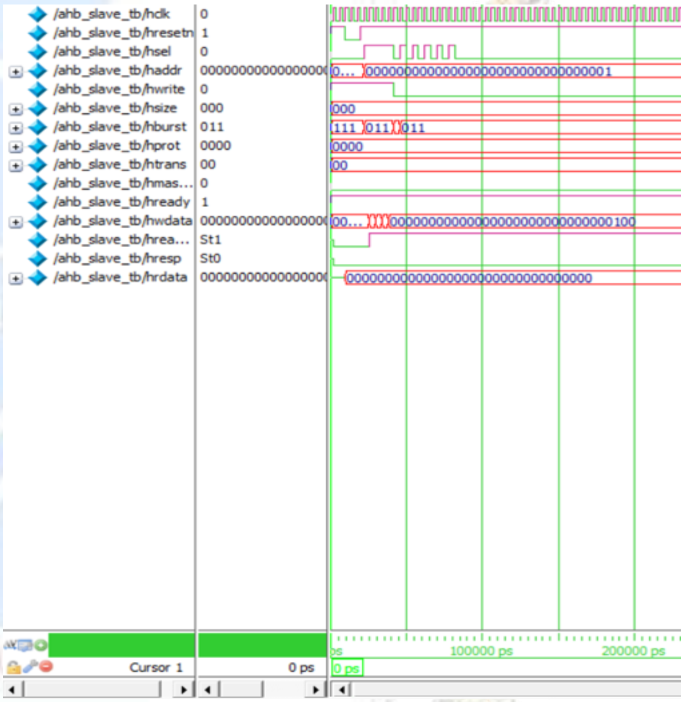


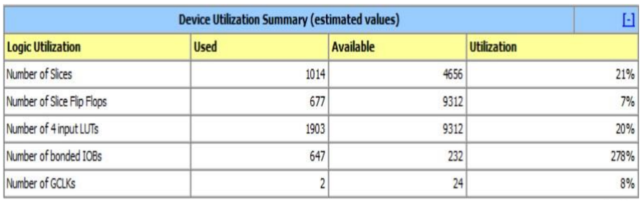
Figure 6 RTL Schematic of AHB Slave



```
/ahb_slave_tb/hclk       0
/ahb_slave_tb/hresetn    1
/ahb_slave_tb/hsel       0
/ahb_slave_tb/haddr      000000000000000000(
/ahb_slave_tb/hwrite     0
/ahb_slave_tb/hsize      000
/ahb_slave_tb/hburst     011
/ahb_slave_tb/hprot      0000
/ahb_slave_tb/htrans     00
/ahb_slave_tb/hmas...    0
/ahb_slave_tb/hready     1
/ahb_slave_tb/hwdata     000000000000000000(
/ahb_slave_tb/hrea...    St1
/ahb_slave_tb/hresp      St0
/ahb_slave_tb/hrdata     000000000000000000(
Cursor 1                 0 ps
```

Figure 7 Simulational Wave of AHB Slave

VII. CONCLUSION: The AMBA AHB circuit design is implemented using with an arbiter and decoder using three masters and with the connection circuit of four slaves, and using and write data multiplexer and read data multiplexer was simulation was done using Xillinx. The simulation report is shown in figure 8.



Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slices	1014	4656	21%
Number of Slice Flip Flops	677	9312	7%
Number of 4 input LUTs	1903	9312	20%
Number of bonded IOBs	647	232	278%
Number of GCLKs	2	24	8%

Fig 8 Simulation Report

VIII. REFERENCES :

[1] S Ramagundam et al. "AMBA AHB ACE and Slave with Memory Controller Interface: Plan and Execution Elements." International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 2, 2013, pp. 234-242

[2] Yashdeep Godhal, Krishnendu Chatterjee, Thomas A. Henzinger "Synthesis of AMBA AHB from formal specification", Springer-Verlag, 10 July 2011, Int J Software Tools Technol Transfer (2013),DOI 10.1007/s10009-011-0207-9

[3] Hwang, S. Y., Kang, D. S., Park, H. J., & Jhang, K. S. (2010). Implementation of self-motivated arbitration scheme for the multilayer AHB bus matrix. IEEE Transactions on Consumer Electronics, 56(2), 365-370. doi: 10.1109/TCE.2010.5464139.

[4] Kurmi, R. S., & Somkuwar, A. (2011). Design of AHB Protocol Block for Advanced Microcontrollers. International Journal of Computer Applications, 32(8), 23-29. doi: 10.5120/4436-6002.

[5] Dubois, M., & Savaria, Y. (2005). A generic AHB bus for implementing high-speed locally synchronous islands. In Southeast Conference, Proceedings. IEEE (pp. 11-16).

[6] Shete, P. S., & Oza, S. (2014). Design of an Efficient FSM for an Implementation of AMBA AHB Master. International Journal of Advanced Research in Computer Science and Software Engineering, 4(3), 267-271.

[7] Jayapraveen, D., & Priya, T. G. (2012). Design of memory controller based on AMBA AHB protocol. Elixir Comp. Sci. Engg. 51A, 2, 11115- 11119.

[8] Verilog HDL: A Guide to Digital Design and Synthesis, By Samir Palnitkar.

[9] ARM Ltd. AMBA specification (Rev 5.0)

[10] www.arm.com