



# Multi-AMBA System Processor Interface for High-Performance SoC Integration

<sup>1</sup>Saichitra G B, <sup>2</sup>Ashwini.V,

<sup>1</sup>Mtech Student, <sup>2</sup>Assistant Professor,

<sup>1</sup>Electronics and Communication Engineering,

<sup>1</sup>BMS College of Engineering, Bangalore, India.

**Abstract :** This paper is an outline of creating a multi AMBA system processor interface, which is designed to interconnect various AMBA processors. The main purpose of this interface is to simplify the connection between different AMBA AHB interfaces and external memory devices like RAM and REGISTER, using the high-performance features of the AMBA AHB. This research examines the use of Application Specific Integrated Circuits (ASICs) for integrating processors with functional units into System-On-Chip (SoC), which can run complex applications. It investigates how in ASIC environment processors communicate with their targets via standardized communication protocol across all targets. The paper also discusses problems connected with data throughput and inter-processor/RTL communication in present-day microprocessors and suggests that multiple AMBAs can be used simultaneously to have access to target components they are meant for. It further introduces a proposed arbitration system that manages multiprocessor access; and explores how bulk data access can be optimized while satisfying major ASIC design constraints like speed, low power consumption and efficient area utilization. In regard to these technologies, it was tested using simulation by verification using Verilog, which gave us remarkable outcomes.

**Index Terms -** AMBA AHB protocol; AMBA processor interface; multiprocessor arbitration; Verilog

## I. INTRODUCTION

The VLSI chips in today's digital landscape power a wide range of devices from small toys to high tech gadgets. These chips have not only become complex to accommodate advanced functions but also increasing its size significantly. For example, AMBA processors can reach extremely high clock speeds. To fully use the capabilities of these processors and align other hardware devices with them, an ASIC design must have a highly efficient interface. Additionally, it would be possible to run multiple tasks concurrently by having an interface that supports multiprocessor access, thereby improving general system performance. This study is aimed at developing an interface which allows for concurrent multiprocessor access to minimize flop-to-flop delay, area requirements and energy consumption. The suggested interface makes use of several AMBA AHB side components that control how read and write operations are done on memory storage elements. These components are interconnected using Verilog ensuring seamless integration between the AHB and the FIFO, RAM, Registers etc. The AMBA AHB bus ensures effective connectivity of processors, on-chip memory, off-chip external memory interfaces through this efficient hardware interface targeting areas such as registers, RAMs, FIFOs, UARTs, PCI interfaces or network links. Efficiency and ease of use are ensured by the design process which employs synthesis and simulation techniques through utilizing EDA hardware simulators. Within AMBA-based microcontrollers, there is a typical configuration that consists of a robust system backbone capable of efficiently dealing with external memory bandwidth issues. Acting as the main connection point for the CPU, on-chip memory, and various Direct Memory Access (DMA) devices, this backbone operates as an efficient system. Specifically, this high-performance backbone is represented by AMBA AHB bus. The multiprocessor access to slave devices is described in hardware architecture shown in Figure . This architecture targets high frequency system modules and serves as a basis for a high-performance system design.

This core architecture ensures smooth connectivity among processors, on-chip memories and off-chip external memory interfaces while also exploiting low power peripheral functions available. It allows for effective integration of processors into both on-chip and off-chip memory interfaces designed for low-power peripheral microcell and macrocell functions. These specifications are deliberately designed to enable easy usage within an efficient design flow incorporating synthesis and automated testing methods.

## II. ARCHITECTURE OF THE AMBABASED MICROCONTROLLER

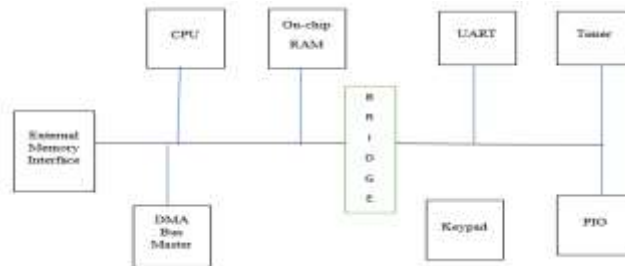


Fig1.AMBA based Microcontroller

A microcontroller built on an AMBA employs a fast system bus that can provide the processor with enough external memory bandwidth it needs, as well as some on-chip memory, UART, Timer and DMA.

For instance, this high-speed bus is part of the AMBA architecture (as seen in Figure 1) which ideally facilitates data transfer between various components through interconnects.

## III. METHODOLOGY

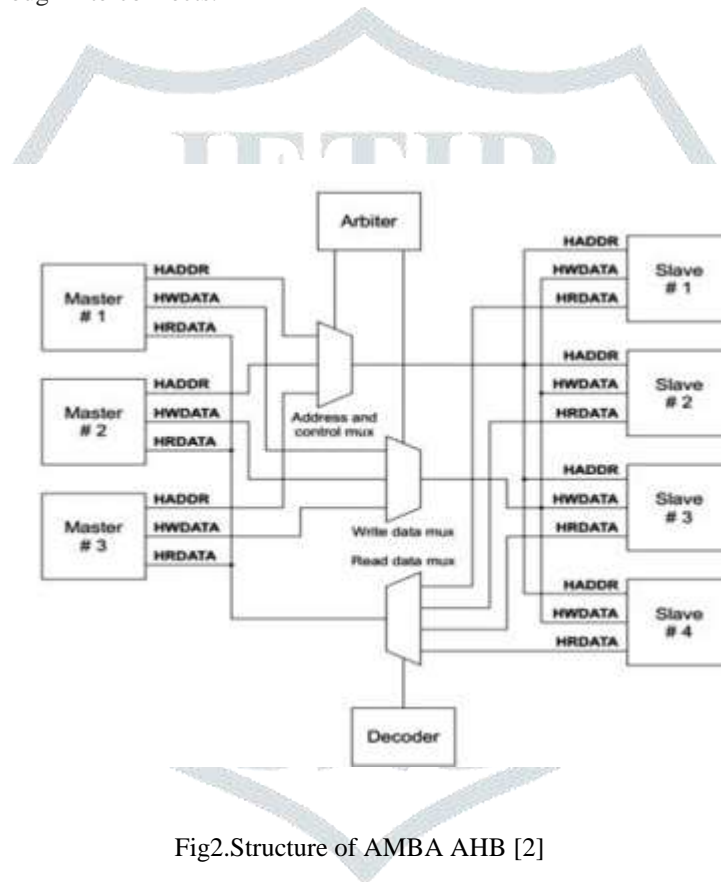


Fig2.Structure of AMBA AHB [2]

### A. AHB MASTER

Master initiates the read and write operations by using address and control information. It ensures the orderly data transfer

### B. AHB SLAVE

Slave responds to the read and write operations initiated by the master within a given address space range. The bus slave signal back to active master about the success and failure or waiting of the data transfer.

### C. AHB ARBITER

AHB Arbiter gives an assurance that only one bus master at a time is allowed to initiate the data transfers.

### D. AHB DECODER

It is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer.

#### IV. HIGH-LEVEL ARCHITECTURE OF MULTIPROCESSOR ACCESS

It provides an interface that allows three AMBA AHB processors to access various target areas depending upon the memory mapping defined by the address bus value of each processor. Some among the target areas include control and status registers, memory, FIFO, UART, PCI interface, and network link. Figure 3 illustrates the interface block diagram interfaced in a multiprocessor access environment.

It is Fig 4 an interface system to a multiprocessor designed for allowing three AMBA AHB processors based on different target areas.

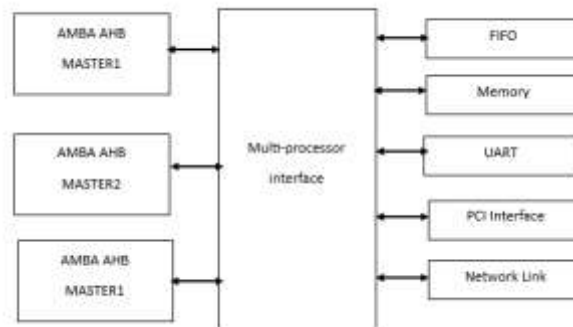


Fig3. AMBA Multiprocessor access Top-level block diagram

- I. **AMBA AHB Masters:** The design incorporates three processors, which in this case, are the AMBA AHB processors. Each of these processors, upon acting as a master, issues requests for access to various target areas.
- II. **Multiprocessor Interface:** This central block serves as the interface, handling the communication between the three AMBA AHB masters and the target areas. It routes the processor requests to the correct target area by memory mapping the address bus values of each processor.
- III. **Target Areas:**
  - A. **Control and Status Registers:** They monitor and control a number of facets of the system.
  - B. **Memory:** General-purpose memory available to all processors for storing and retrieving data.
  - C. **FIFO:** This is a sort of data buffer or queue that allows a program to read data in the order in which it was written.
  - D. **UART:** This is a hardware communication protocol adapted to be used in asynchronous serial communications from one device to another.
  - E. **PCI Interface:** A connector interface that links high-speed peripherals such as network cards, amongst others.
  - F. **Network Link:** This acts to provide network connectivity, thus allowing the processors to intercommunicate over a network.

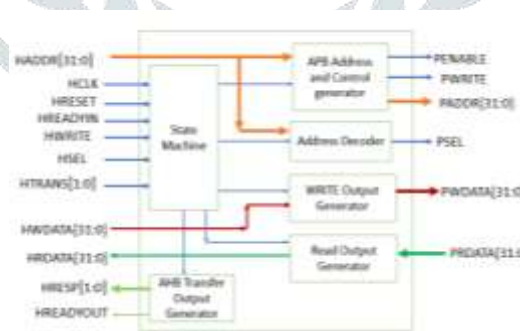


Fig4. Multiprocessor interface

##### 1. State Machine:

This block controls the time of activity of various output signals are active. This block contains the two most critical signals current state and next state three bit signals. For the generation of next state, more signals are required as HWRITE and the registered version of HWRITE(Reg\_write), accept and the current state.

2. **Write output Generator:** This sub-block uses a DFF to control the flow of HCLK, asynchronous reset signal, HRESET and a data input, HWDATA. The output of the flip flop, PWDATA, duplicates the input data with regard to the clock signal and also includes an asynchronous reset of the output.

3. **Read output Generator:** To design the necessary logic with two D Flip/Flops (DFFs), which will be accomplished by two stages of DFFs' first DFF will take HCLK, HRESET and HWDATA as input and its drive PWDATA as the output. The second DFF will take HCLK, HRESET, and PRDATA as inputs and drive HRDATA as the output. Both DFFs will have an asynchronous reset.

4. *AHB Transfer output Generator*: In order to add transfer generation, response signals and a state machine, we will design a module generating HREADYOUT and HRESP signals dependent on condition of a basic state machine and the AHB inputs. The HRESP signal this will always be set to 00 to indicate a successful transfer.

## V.ARBITER

An AMBA AHB bus arbiter guarantees that, at any point in time, only one bus master is allowed to initiate the transfer of data. This is necessary for preventing conflicts on the bus. This is made possible by a Round Robin algorithm; the actual algorithm used might be of the highest priority or fair access or fixed priority depending on application needs.

### A. ARBITRATION SYSTEM:

In that case, an arbitration mechanism is of essence in ensuring order and conflict-free access to the bus within an Advanced High-performance Bus system. The simple role of an arbiter is essentially to ensure that only one bus master is able to have control over the bus at any particular time. This is achieved through an evaluation of the requests from multiple bus masters and selecting the highest-priority master to be granted access to the bus. The arbiter also handles, furthermore, the requests from the slaves involved in SPLIT transfers.

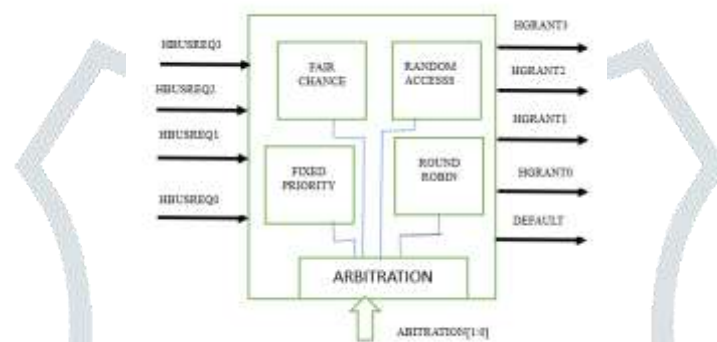


Fig5.Functional diagram of AHB Arbiter

#### i. Arbitration Modes:

- Fair Chance:** In this type, each requesting device will have an equal chance of accessing the bus, perhaps through proportional division of access.
- Random Access:** This mode provides random access to the bus. Such random access may be appropriate in some cases where access unpredictability is a desirable feature.
- Fixed Priority:** In this method, priorities are assigned to various requests. It may be set up to accord greater priority to certain devices on predefined criteria.
- Round Robin:** Here, bus access is provided in a cyclical order. That is to say that each device will have its turn against a fixed sequence.

#### ii. Arbitration Select Controller:

- Arbitration Select Controller:** This controller chooses the arbitration mode based on the arbitration select signal (ARBITRATION [1:0]).
- Arbitration [1:0]:** These are the select signals used to select the type of arbitration to be used.

#### iii. Request and Grant Signals:

- HBUSREQ<sub>3</sub> to HBUSREQ<sub>0</sub>:** These signals represent bus request inputs from four different sources; for example, devices or modules requesting access to the bus.
- HGRANT<sub>3</sub> to HGRANT<sub>0</sub>:** These signals constitute the bus grant outputs, thus defining from which requesting source the bus should be granted for one's use.
- DEFAULT:** This could either be used with respect to handling situations where none of the specific requests get granted or some other form of fallback mechanism.

A. State Machine for Multi-Mode Bus Arbitration Control

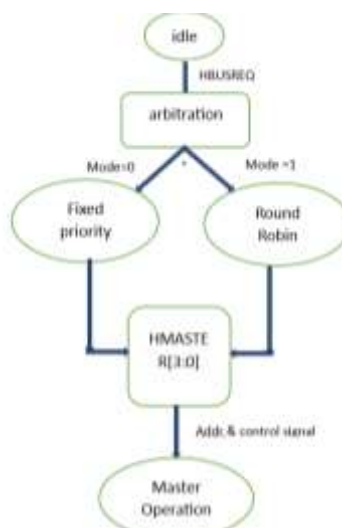


Fig6. State diagram of AHB arbiter

VI.RESULTS AND DISCUSSION

The proposed system interfaces with four master AMBA AHB buses. In this paper, we have tested these input signals using Verilog. An arbitration system was introduced for the access of multiprocessors to ensure that multiple AMBA processors could access their targets simultaneously without any conflict. It efficiently optimized the access of bulk data while keeping the constraints of low power and efficient area usage, which are the key constraints in ASIC design. Implementation of the target area FIFO is carried out and interfaced with AHB through processor and results are verified. Fig.7 Fig.8 shows the wave form of simulation of the AHB and FIFO design, respectively.

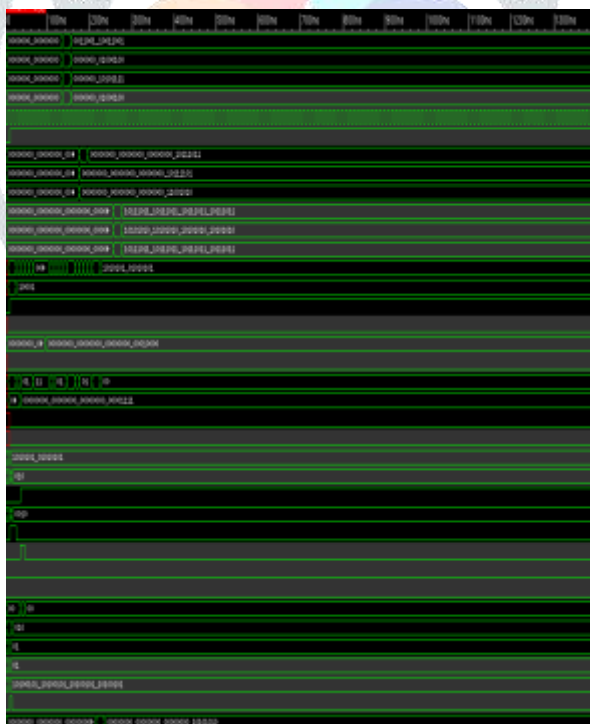


Fig7. Simulation of AMBA AHB

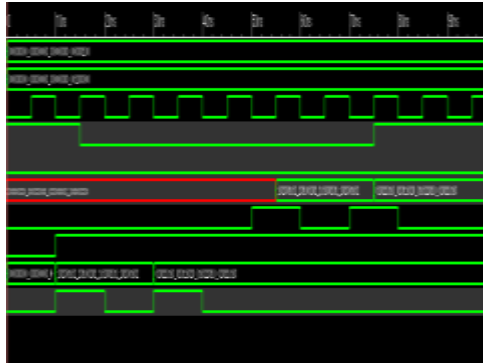


Fig 9. Simulation waveform of FIFO

## VII.CONCLUSION

The study presents the design of a robust multi-AMBA system processor interface for efficient communication between the AMBA AHB processor and the external memory units. This makes use of ASIC integration and Verilog simulation to solve research challenges on data throughput and inter-processor communication. It proposes the arbitration system and the optimization strategies, underpinning the possibility of attaining efficient SoC designs that have a focus on speed, reducing power consumption, and efficient area usage. This work contributes to important insights in the development of high-performance System-On-Chip architectures that pave the way for further progress in the execution of complex applications and multiple processor integration.

## VIII.REFERENCES

- [1] S.M.A. Motakabber, Mohammad Abdulmatin and M.D.Mahmud " Advancing System Integration: Verilog - Based Hardware Implementation of an ASIC Interface for Three AMBA Processors" IIUM Engineering Journal, Vol. 25, No. 1 june, 2024.
- [2] Saluja H, Grover N. "Multiple Master Communication in AHB IP using Arbiter". International Journal of Engineering and Manufacturing (IJEM), 10(1): 2020.
- [3] Mishra A. "Design of a Round Robin Bus Arbiter using System Verilog". International Research Journal of Engineering and Technology (IRJET), 7(1): 2020.
- [4] Giridhar P, Choudhury P. (2019) Design and Verification of AMBA AHB. 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), 19-20th Mar 2019.
- [5] Deekhsa L, Shivakumar BR. (2019) Effective Design and Implementation of AMBA AHB Bus Protocol using Verilog. International Conference on Intelligent Sustainable Systems (ICISS), 21- 22nd Feb 2019, Palladam, India
- [6] Tu J, Ou C. (2017) A practice of 2-to-2 fixed priority arbiter used to multi-core processors. International Conference on Applied System Innovation (ICASI 2017), 13-17th May 2017, Sapporo, Japan
- [7] S. Ramagundam et al., "Design and implementation of highperformance master/slave memory controller with microcontroller bus architecture," Conf. Rec. - IEEE Instrum. Meas. Technol. Conf., no. May, pp. 10–15, 2014.
- [8] Iqbalur R. and Reasat C. (2011) Hardware Implementation of StrongARM Processor Interface Using Verilog & FPGA. International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2011). Pattaya, Thailand, Dec 17-18, 2011
- [9] Dobberpuhl DW. (1997) Circuits and technology for Digital's StrongARM and ALPHA microprocessors [CMOS technology]. In Proceedings Seventeenth Conference on Advanced Research in VLSI, 2-11. doi: 10.1109/ARVLSI.1997.634842.
- [10] M. K. Kumar, A. Sajja, and F. Noorbasha, "Design and FPGA Implementation of AMBA APB Bridge with Clock Skew Minimization Technique," vol. 7, no. 3, pp. 42–45, 2017.