

ADVANCING SYSTEM INTEGRATION: VERILOG-BASED HARDWARE IMPLEMENTATION OF AN ASIC INTERFACE FOR THREE AMBA PROCESSORS

S.M.A. MOTAKABBER^{1*}, MOHAMMAD IQBAL RAHMAN ROKON¹,
AHM ZAHIRUL ALAM¹, GAZI ZAHIRUL ISLAM¹,
MOHAMMAD ABDULMATIN² AND MD. MAHMUD³

¹*Department of Electrical and Computer Engineering,
International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia*

²*Department of Electrical and Computer Engineering, North South University,
Dhaka, Bangladesh*

³*Department of Electrical and Computer Engineering, College of Engineering,
Lamar University, Beaumont-77710, Texas, USA*

*Corresponding author: amotakabber@iium.edu.my

(Received: 13 June 2023; Accepted: 12 November 2023; Published on-line: 1 January 2024)

ABSTRACT: This paper presents the development of a multi-AMBA system processor interface employing multiple AMBA processors. The primary goal of this interface is to establish connections between various AMBA AHB interfaces and external memory units such as RAM and REGISTER, leveraging the high-performance capabilities of AMBA AHB. The research delves into the utilization of ASICs to integrate processors and functional blocks into a System-On-Chip (SoC) configuration, enabling the execution of intricate applications. Within the ASIC environment, the research explores how processors communicate with their designated targets through an interface that standardizes the communication protocol for all targets. It underscores the challenges posed by data throughput and inter-processor/RTL communication in contemporary processors and suggests the concurrent use of multiple AMBA processors for accessing their respective targets. Additionally, the paper introduces an arbitration system for managing multiprocessor access and investigates the optimization of bulk data access while prioritizing crucial ASIC design constraints, including speed, low power consumption, and efficient area utilization. The proposed system was rigorously validated through simulation using Verilog HDL, yielding positive and promising results.

ABSTRAK: Kajian ini adalah mengenai pembangunan antara muka, sistem pemproses berbilang AMBA yang mengandungi berbilang pemproses AMBA. Tujuan antara muka ini adalah bagi mewujudkan hubungan pelbagai antara muka AMBA AHB dengan unit memori luaran seperti RAM dan REGISTER, ini sekaligus memanfaatkan keupayaan tinggi AMBA AHB. Kajian ini mengguna pakai ASIC bagi menyatukan pemproses dan blok berfungsi pada konfigurasi Sistem-Atas-Cip (SoC), membolehkan pelaksanaan aplikasi rumit. Pada persekitaran ASIC, kajian ini meneroka cara pemproses berkomunikasi dengan sasaran yang ditetapkan melalui perantaraan antara muka yang menyeragam protokol komunikasi bagi semua sasaran. Ia menggariskan cabaran yang ditimbulkan oleh pemprosesan data dan komunikasi antara pemproses/RTL dalam pemproses kontemporari dan mencadangkan penggunaan secara serentak pemproses berbilang AMBA bagi mengakses sasaran masing-masing. Selain itu, kertas kerja ini memperkenalkan sistem timbang tara bagi mengurus akses berbilang pemproses dan mengkaji akses data pukal yang optimum sambil mengutamakan kekangan reka bentuk ASIC, seperti kelajuan, penggunaan kuasa rendah dan penggunaan

kawasan secara cekap. Sistem ini telah disahkan dengan teliti melalui simulasi menggunakan Verilog HDL, memberikan dapatan positif dan harapan baik.

KEYWORDS: *AMBA protocol; AMBA processor interface; multiprocessor arbitration; FPGA; Verilog HDL*

1. INTRODUCTION

In the modern era, digital control is pervasive, extending from simple toys to sophisticated high-tech devices, all orchestrated by VLSI chips [1]. These chips have not only become increasingly complex to accommodate advanced functions but have also grown significantly in size. Devices like the AMBA processor can operate at extremely high clock speeds. To harness the full potential of these processors and synchronize the remaining hardware with their speed, an efficient interface is imperative for any ASIC. Furthermore, if this interface can support multiprocessor access instead of a single processor, it would enable concurrent execution of tasks. This research aims to address these critical challenges and propose an interface capable of concurrent multiprocessor access with reduced flop-to-flop delay, minimized area requirements, and lower power consumption. The interface is constructed using multiple AMBA AHB-side components for controlling read and write operations on memory storage elements from the AHB side. All of these components are interconnected using Hardware Description Language (HDL), ensuring a seamless connection between the AHB Bridge, RAM, and registers. Using the efficient hardware interface, AMBA AHB bus ensures the efficient connection of processors, on-chip memory and off-chip external memory interfaces with targeted areas such as be registers, RAMs, FIFO, UART, PCI Interface or Network link. It also ensures ease of use in an efficient design flow using synthesis and simulation techniques using EDA hardware simulator [1].

When it comes to AMBA-based microcontrollers, the standard configuration includes a sturdy system backbone that can efficiently manage external memory bandwidth. This backbone serves as the central hub connecting the CPU, on-chip memory, and various Direct Memory Access (DMA) devices. More specifically, the AMBA AHB bus functions as this high-performance system backbone. The hardware architecture of multiprocessors access to the slave devices [2-4] is shown in Fig. 1. It is finely tuned for use in high-frequency system modules, forming the core of the high-performance system architecture. This core ensures smooth connectivity between processors, on-chip memories, and off-chip external memory interfaces, all while accommodating low-power peripheral microcell functions. Moreover, it facilitates the efficient connection of processors, on-chip memory, and off-chip external memory interfaces to support low-power peripheral macro cell functions. Its specifications are carefully crafted to guarantee ease of use within an efficient design process, incorporating synthesis and automated testing techniques.

2. METHODOLOGY

This research followed the Top-Down Hierarchical Technique to partition the design. Processor interfaces provide bridge connection between processor and rest of the hardware. Processors operate as per its bus protocols: it can be processor specific or standard protocol. Interface needs to understand those protocols and provide access into the targets like registrar, memory, FIFO, network link or other PCI interfaces as per their own access control. This research will focus on developing an interface that can support multiple AMBA AHB processor access concurrently with higher speed, lower power and area. As designing multi processors are complex both in hardware and software, so in our research we need to focus on managing

complexity using hierarchical methodology and we need to be further concern if one processor fails then it will not affect in the speed. It should perform in low power and should take low area. After design specification is complete, interface will be modeled using Verilog HDL. Below is the Verilog HDL based ASIC design flow.

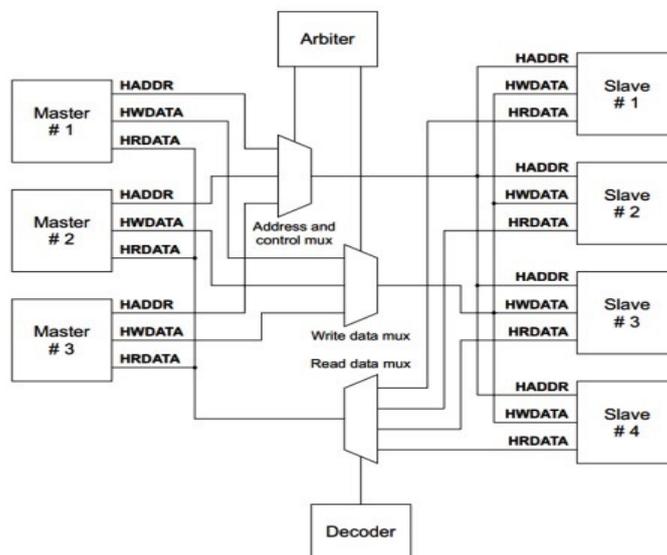


Fig. 1: Multiplexer interconnections for multiple AHB masters [2-4].

2.1 AHB Signals and Transfers Involving AMBA Masters

2.1.1 HADDR [31:0]

During write operations, the write data bus plays a critical role in conveying data from the master to the bus slaves. It is highly recommended to maintain a minimum data bus width of 32 bits. Nonetheless, it can be effortlessly expanded to facilitate higher bandwidth [3,4].

2.1.2 HWDATA [31:0]

The write data bus is driven by the bus master during write transfers. If the transfer is extended then the bus master must hold the data valid until the transfer complete, as indicated by HREADY HIGH.

2.1.3 HRDATA [31:0]

During read operations, the relevant slave device takes control of the read data bus. If this slave prolongs the read process by keeping the signal HREADY in a LOW state, it is only required to furnish valid data at the conclusion of the final cycle of the transfer. This point is signified by the transition of HREADY to a HIGH state, as documented in reference [3].

3. RTL HARDWARE DESIGN SPECIFICATION

Design specification enables designers to implement their ideas systematically on paper. This specification appears in text/document format and represent what designers expect their chip to do. At this point of the design process, designers demonstrate very little concern about hardware consideration such as area, speed or power consumed by the chip. Design specification contains multiple identical processors (AMBA processor in our case). It is capable of interpreting processor BUS protocols in order to read/write operations on target devices.

Both Fixed and Round-Robin arbitration systems would be available to deal with situations requiring multiple processors to access a particular generic device concurrently [5].

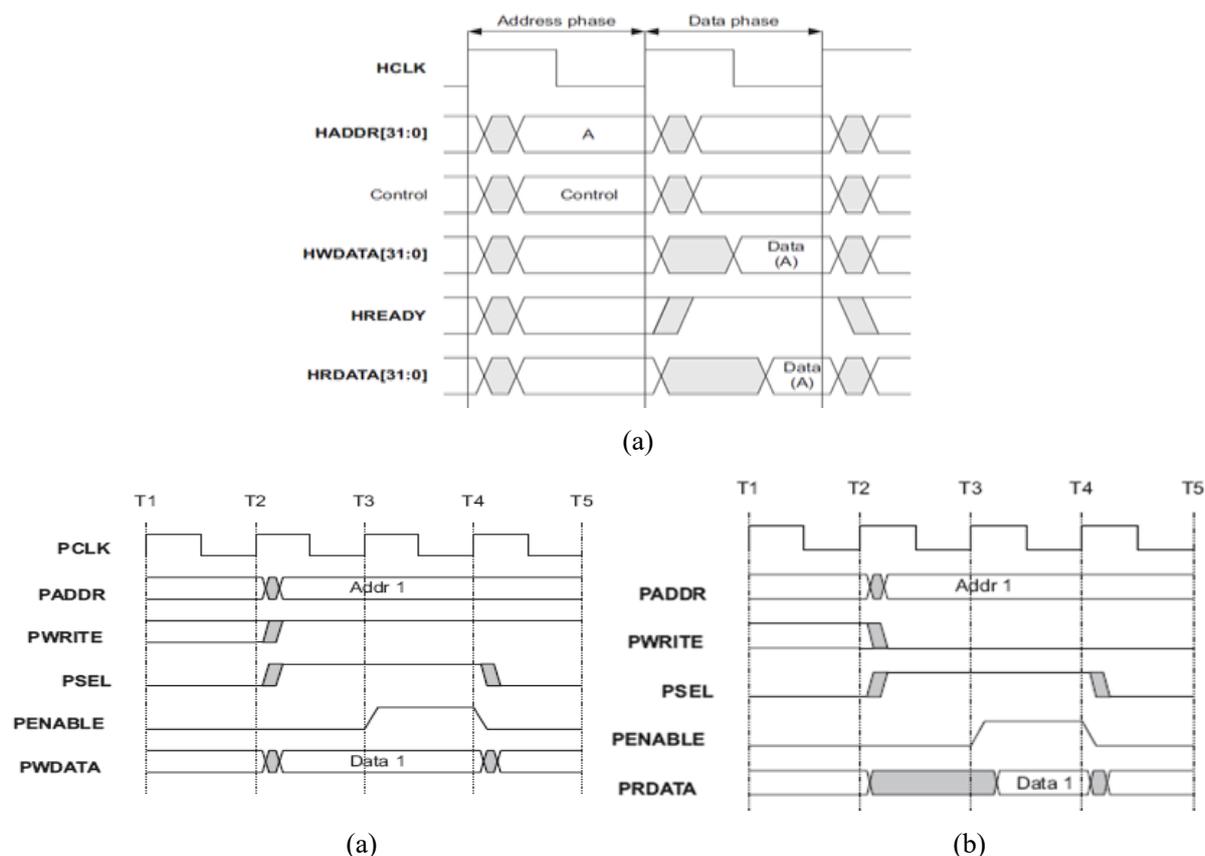


Fig. 2: AMBA transfer (a) simple with no wait states, (b) write and (c) read.

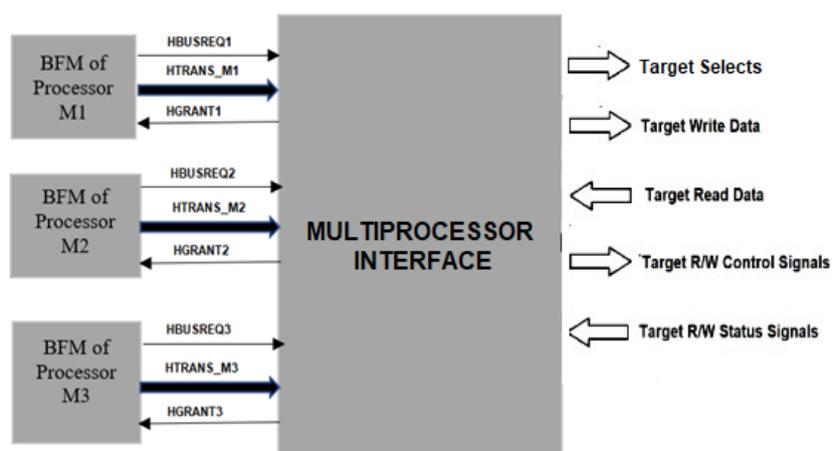


Fig. 3: Proposed simulation test environment.

Control flow and simulation test environment for the proposed AMBA multiprocessor interface is shown in Fig. 3.

In this design specification we are using four AMBA Master AHB bus and three input signals which are HADDR, HWDATA and HRDATA. These signals will go to the slaves which can be RAM, REGISTER, and FIFO etc. These three master signals cannot reach the

slaves altogether at a time. So, for this arbiter will select which signal will reach the slaves first. There are address and control mux, write data mux, read data mux and decoder. This design specification is implemented for 32-bit, 64 bit and 128 bits. The address signal HADDR will address the signal. If the arbiter selects 00 then it will select HADDR1, for 01 it is HADDR2, for 10 it is HADDR3 and for 11 it is HADDR4 like this [6]. Same thing will be applicable for other signals [7].

3.1 Top Level Block Diagram of Multiprocessor Access

Interface allows three AMBA AHB processors to access the target areas according memory mapping of each processor address bus value. Target areas can be Control and Status Registers, Memory, FIFO, UART, PCI Interface and Network Link. Figure 4 shows the block diagram of interface in multiprocessor access [2,6].

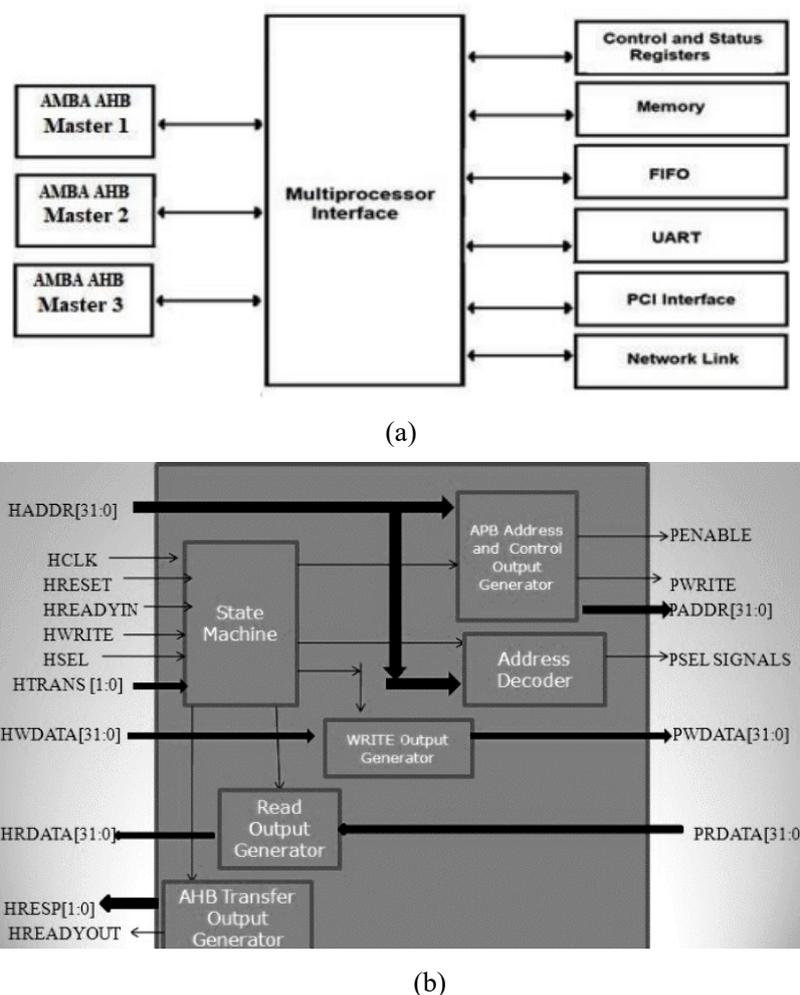


Fig. 4: (a) Top level block diagram in AMBA multiprocessor access, and (b) major blocks inside the interface [2,6].

3.2 State Machine

This is a very vital block for the interface. It determines when different output signals will come in effect. Two most significant signal of this block is next state and current state both of which are 3 bit signals. To generate next state some other signals are needed like HWRITE, registered version of HWRITE (Reg. write), accept and current state as well. Next state and current state will be used as internal input signals in other blocks [7].

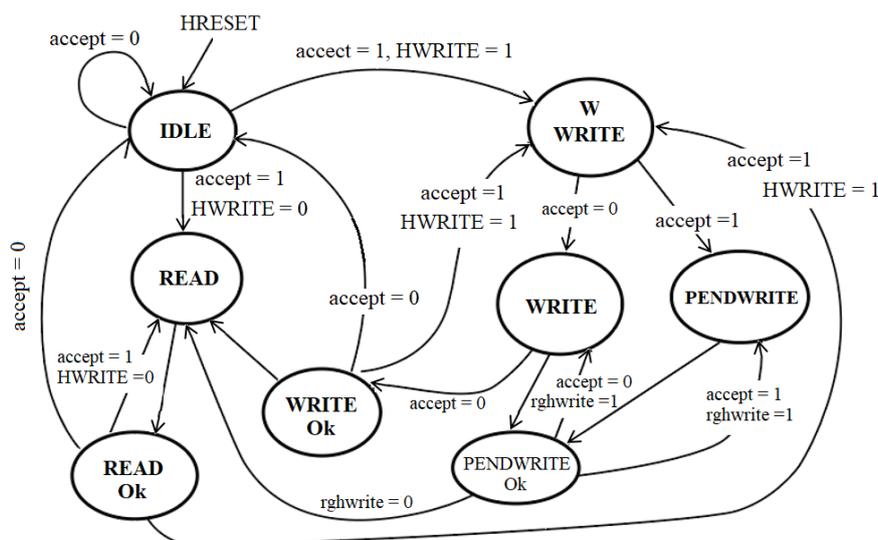


Fig. 5: Processor access control state machine.

3.2.1 Write Output Generator

This sub-block is implemented by a DFF. Here input will be taken as HCLK, HRESET and HWDATA then output will be driven to PWDATA. Here reset is asynchronous.

3.2.2 Read Output Generator

To implement this one more DFF is used. Here input will be taken as HCLK, HRESET and PRDATA. The output will then drive to HRDATA. Asynchronous reset is used.

3.2.3 AHB Transfer Output Generator

Here transfer generation is used as output HREADYOUT and HRESP as response signals. State machine and AHB inputs generates HREADYOUT. To show the complete of transfer 00 is always set to HRESP.

3.3 AHB Arbiter

The bus arbiter guarantees that only one bus master can initiate data transfers at any given time. Despite having a fixed arbitration protocol, various arbitration algorithms, such as highest priority or fair access, can be implemented based on the specific requirements of the application. An AHB system typically incorporates only one arbiter, although this is a straightforward setup in single bus master systems. The AHB decoder is responsible for decoding the address of each data transfer and generating a select signal for the slave involved in the transfer. In all AHB implementations, a single centralized decoder is necessary [4].

3.3.1 Arbitration System

The arbitration mechanism's primary purpose is to uphold the principle of allowing only one master to utilize the bus at a time. This function is executed by the arbiter, which assesses multiple requests from potential bus users and identifies the current highest-priority master seeking bus access. Furthermore, the arbiter handles requests from slaves wishing to execute SPLIT transfers. For slaves not engaged in SPLIT transfers, the intricacies of the arbitration process may not be relevant, but it's essential for them to acknowledge that a series of transfers could remain incomplete if the bus's ownership status changes [8].

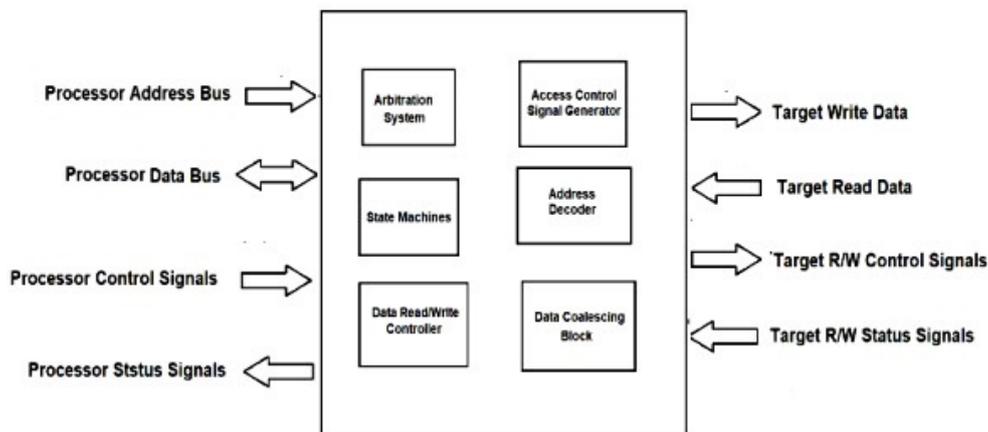


Fig. 6: Arbitration system inside the interface [9].

3.3.2 Arbitration Control State Machine

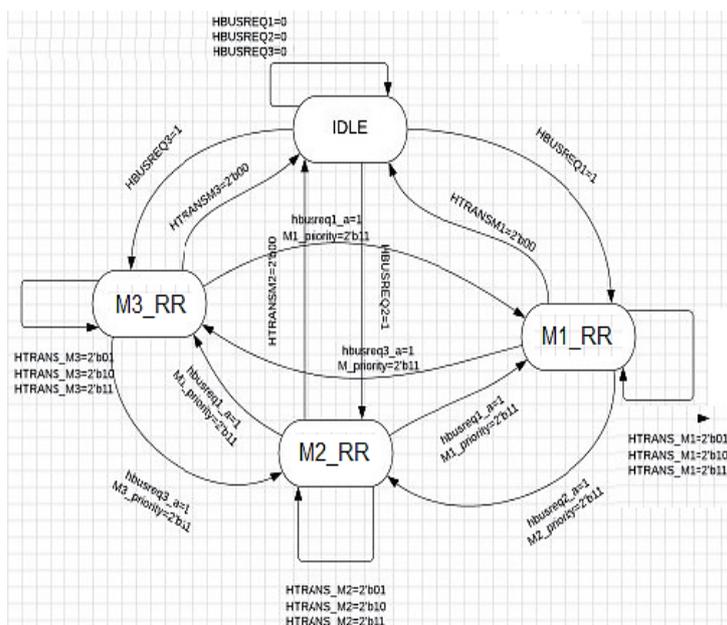


Fig. 7: State transitions of dual mode arbitration under Round Robin.

4. RESULTS AND DISCUSSION

The proposed system has implemented the interface by design using four master AMBA AHB buses and each contains three input signals HADDR, HWDATA and HRDATA. We tested the input signals in Verilog HDL. In our result we got some signals of high impedance which were not expected. But the other outputs of the timing diagram were according to our desired output. This work is extended to simulation part. And further work will cover the simulation of the designed part. After implementing design improvements, simulation results will be studied and timing diagram will be analyzed and compared with results. Efficient usage of logic synthesis tools will limit area requirements and power consumption of transistors inside chip as per requirements of the design. The wave form of simulation of the design is shown in Fig. 8.

The synthesis reports are shown in Table 1. Analyzing FSM < FSM_0 > for best encoding. Optimizing FSM < FSM_0 > on signal < currentstate [1:3] > with gray encoding.

Table 1: Logic state, and list of components and their numbers

State	Encoding	Macro Statistics	Macro Structure	No. of blocks
000	000			
010	001	Registers : 1	Registers : 60	
001	011	Flip-Flops : 1	1-bit register : 20	
101	010	Multiplexers : 303	32-bit register : 40	
100	110	32-to-1 multiplexer : 1303	Multiplexers : 1	
011	111		32-to-1 multiplexer : 1	
111	101			
110	100			

While previous research has explored multiprocessor system interfaces, our work takes a unique approach by focusing on multi AMBA AHB system processors, which has not been extensively addressed before [5]. Unlike prior studies that concentrated on a single signal, we have developed our interface design to accommodate four AMBA AHB bus protocols, effectively serving as masters for these protocols. These protocols utilize three crucial input signals: HADDR, HWDATA, and HRDATA.

Furthermore, considering the ever-evolving landscape of technology, we have future-proofed our design to support not only the current 32-bit standard but also 64-bit and even 128-bit configurations [9]. Our upcoming plans involve delving into system coalescence to enable efficient bulk data access. We aim to implement two distinct arbitration systems, namely Fixed and Round-Robin, which will come into play when multiple processors simultaneously require access to the same location. This forward-looking approach positions our research to make a meaningful contribution to the future of processor technology.

5. CONCLUSION

This study aims to develop a cutting-edge System-On-Chip (SoC) with a high-speed processor by introducing an efficient integrated interface. The primary goal of this research is to create an interface that supports multiple AMBA AHB system processors. The project involves the design of multiple AHB Masters within a multi-AMBA system using Verilog HDL. Building upon the concepts presented in the references, this work delves into the intricacies of multiprocessor concurrent access, including the implementation of advanced arbitration mechanisms and coalescing techniques. The anticipated result of this research is the potential to contribute to the development of future high-tech chips with intricate functionalities operating at top speeds.

REFERENCES

- [1] 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, India.
- [2] Iqbalur Rahman Rokon TRAA. (2011) Hardware Implementation of AMBA Processor, in International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2011), Pattaya, Dec 17-18, 2011.
- [3] 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.

- [4] AMBA: The Standard for On-Chip Communication, [Online]. Available: <https://www.arm.com/products/siliconip-system/embedded-system-design/amba-specifications>.
- [5] 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.
- [6] Saluja H, Grover N. (2020) Multiple Master Communication in AHB IP using Arbiter. *International Journal of Engineering and Manufacturing (IJEM)*, 10(1): 29-40.
- [7] 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.
- [8] 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.
- [9] Mishra A. (2020) Design of a Round Robin Bus Arbiter using System Verilog. *International Research Journal of Engineering and Technology (IRJET)*, 7(1): 29-40.