International Journal of Engineering Research and Reviews ISSN 2348-697X (Online)
Vol. 2, Issue 3, pp: (56-60), Month: July - September 2014, Available at: www.researchpublish.com

Slave-Side Arbitration and Implementation For Multilayer - AHB Bus Matrix

[1]Mrs.R.Ramya, [2]Ms.Preethi Vadivel

1, 2 Assistant Professor, Department of Electronics and Communication Engineering, Gnanamani College of Technology, Namakkal, India

Abstract: **The multilayer advanced high-performance bus (ML-AHB) bus matrix employs slave-side arbitration. Slave-side arbitration is different from master-side arbitration in terms of request and grant signals since, in the former, the master merely starts a burst transaction and waits for the slave response to proceed to the next transfer. Therefore, in the former, the unit of arbitration can be a transaction or a transfer. However, the ML-AHB bus matrix of ARM offers only transfer-based fixed-priority and round-robin arbitration schemes.**

In this paper, the design and implementation of a flexible arbiter for the ML-AHB bus matrix to support three priority policies—fixed priority, round robin, and dynamic priority and three data multiplexing modes transfer, transaction, and desired transfer length. In total, there are nine possible arbitration schemes. The proposed arbiter, which is self-motivated (SM), selects one of the nine possible arbitration schemes based upon the priority-level notifications and the desired transfer length from the masters so that arbitration leads to the maximum performance.

Keywords: **AHB Bus Matrix, Slave-Side Arbitration, High-performance, ML-AHB bus matrix.**

I. INTRODUCTION

The ON-CHIP bus plays a key role in the system-on-a-chip (SoC) design by enabling the efficient integration of heterogeneous system components such as CPUs, DSPs, application- Specific cores, memories, and custom logic. Recently, as the level of design complexity has become higher, SoC designs require 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 the multilayer AHB (ML-AHB) bus matrix from ARM, the PLB crossbar switches from IBM, and CONMAX from Silicore. Among them, the ML-AHB bus matrix 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 systems with low power. The ML-AHB bus matrix 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. In particular, the ML-AHB bus matrix uses slave-side arbitration. Slave-side arbitration is different from master-side arbitration in terms of request and grant signals since, in the former, the master merely starts a burst transaction and waits for the slave response to proceed to the next transfer. Therefore, the unit of arbitration can be a transaction or a transfer. The transaction-based arbiter multiplexes the data transfer based on the burst transaction, and the transfer-based arbiter switches the data transfer based on a single transfer. However, the ML-AHB bus matrix of ARM presents only transfer-based arbitration schemes, i.e., transfer based fixed-priority and round-robin arbitration schemes.

Hence, our arbiter is able to not only deal with the transfer-based fixed-priority, round-robin, and dynamic-priority arbitration schemes but also manage the transaction-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. Furthermore, our arbiter provides the desired-transfer-length-based-fixed-priority, round-robin, and dynamic-priority arbitration schemes. In addition, the proposed SM arbiter selects one of the nine possible arbitration schemes based on the priority-level notifications and the desired transfer length from the masters to ensure that the arbitration leads to the maximum performance.

II. PROPOSED SLAVE-SIDE ARBITRATION FOR ML-AHB BUS MATRIX

In contrast, it is possible to deal with the transfer-based arbitration scheme as well as the transaction-based arbitration scheme in slave-side arbitration. In this paper, we propose a flexible arbiter based on the self-motivated (SM) arbitration scheme for the ML-AHB bus matrix. Our SM arbitration scheme has the following advantages:

1) It can adjust the processed data unit;

2) It changes the priority policies during runtime; and

3) It is easy to tune the arbitration scheme according to the characteristics of the target application.

Hence, our arbiter is able to not only deal with the transfer-based fixed-priority, round-robin, and dynamic-priority arbitration schemes but also manage the transaction-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. Furthermore, our arbiter provides the desired-transfer-length-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. In addition, the proposed SM arbiter selects one of the nine possible arbitration schemes based on the priority-level notifications and the desired transfer length from the masters to ensure that the arbitration leads to the maximum performance.

The ML-AHB busmatrix of ARM consists of the input stage, decoder, and output stage, including an arbiter. The input stage is responsible for holding the address and control information when transfer to a slave is not able to commence immediately. The decoder determines which slave that a transfer is destined for. The output stage is used to select which of the various master input ports is routed to the slave. Each output stage has an arbiter. The arbiter determines which input stage has to perform a transfer to the slave and decides which the highest priority is currently. The ML-AHB busmatrix employs slave-side arbitration, in which the arbiters are located in front of each slave port, the master simply starts a transaction and waits for the slave response to proceed to the next transfer.

III. ARBITRATION SCHEMES FOR THE ML-AHB BUS MATRIX OF ARM

The ML-AHB bus matrix of ARM consists of the input stage, decoder, and output stage, including an arbiter. Fig. Shows the overall structure of the ML-AHB bus matrix of ARM. The input stage is responsible for holding the address and control information when transfer to a slave is not able to com-mence immediately. The decoder determines which slave that a transfer is destined for. The output stage is used to select which of the various master input ports is routed to the slave.

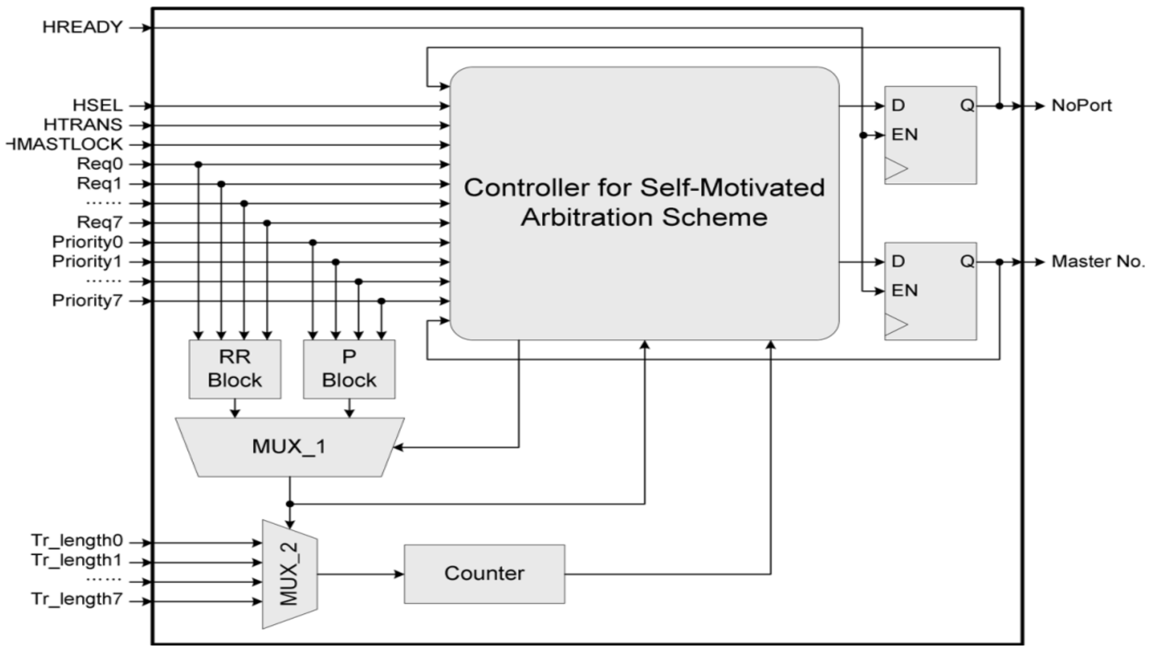

Fig-1 Internal structure of our arbiter

The arbiter determines which input stage has to perform a transfer to the slave and decides which the highest priority is currently. The ML-AHB bus matrix em-ploys slave-side arbitration, in which the arbiters are located in front of each slave port, as shown in Fig. 1; the master simply starts a transaction and waits for the slave response to proceed to the next transfer. Therefore, the unit of arbitration can be a transaction or a transfer. However, the ML-AHB bus matrix of ARM furnishes only transfer-based arbitration schemes, specifically transfer-based fixed-priority and round-robin arbitration schemes. The transfer-based fixed-priority (round-robin) arbiter multiplexes the data transfer based on a single transfer in a fixed-priority or round-robin fashion.

IV. AMBA BUSES

The **Advanced Microcontroller Bus Architecture (AMBA)** specification defines an on-chip communications standard for designing high-performance embedded microcontrollers.

Three distinct buses are defined within the AMBA specification:

- the Advanced High-performance Bus (AHB)
- the Advanced System Bus (ASB)
- the Advanced Peripheral Bus (APB).

THE AMBA AHB

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation.

AMBA AHB implements the features required for high-performance, high clock frequency systems including:

- burst transfers
- split transactions
- single-cycle bus master handover
- single-clock edge operation
- non-tristate implementation

Wider data bus configurations (64/128 bits).

A typical AMBA AHB system design contains the following components:

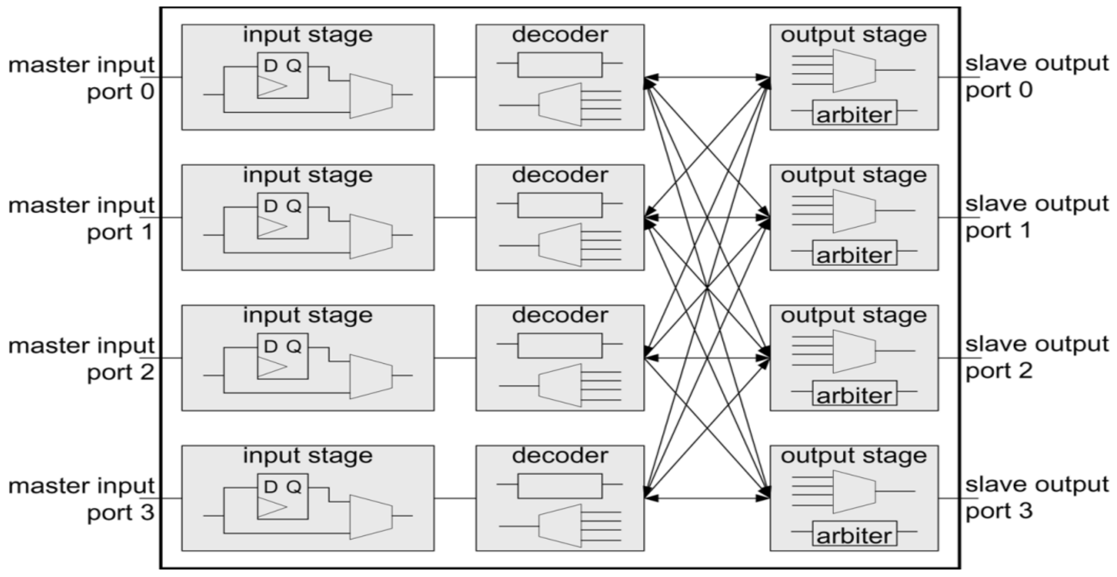

Fig-2 Structure of ML-AHB bus matrix of ARM

AHB master-A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.

AHB slave-A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer**.**

AHB arbiter-The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as *highest priority* or *fair* access can be implemented depending on the application requirements. An AHB would include only one arbiter, although this would be trivial in single bus master systems.

AHB decoder -The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementation

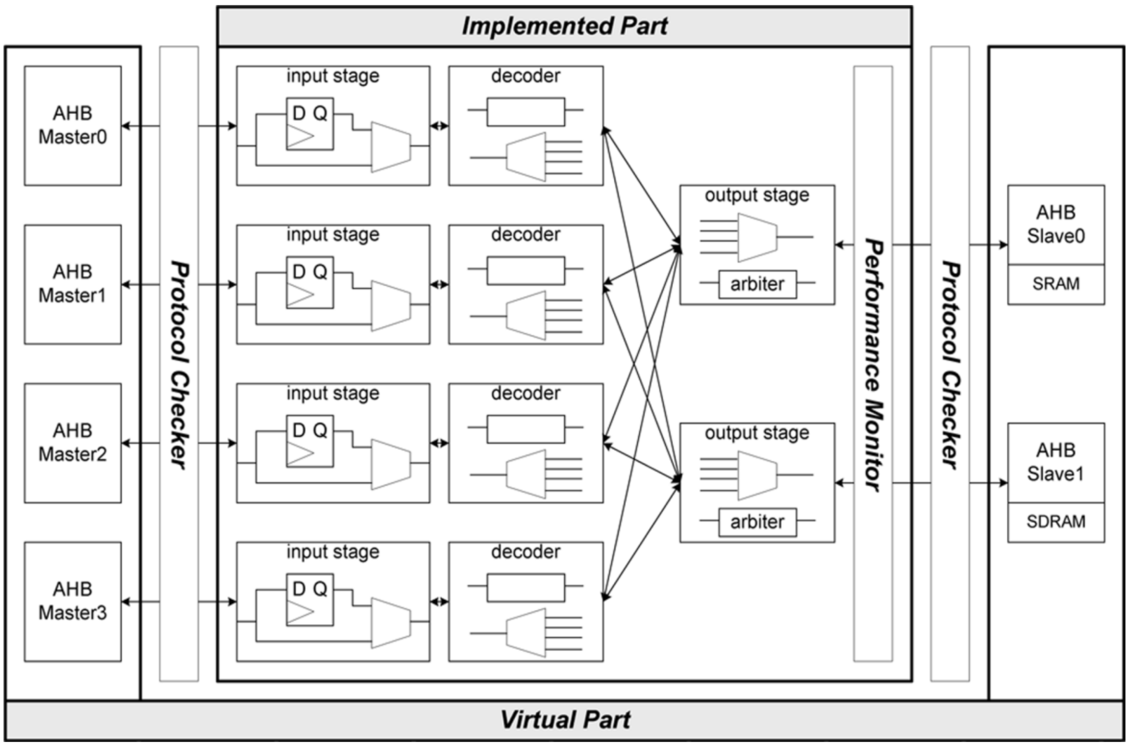

Fig-3 Simulation environment for performance analysis

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 [1:0]**:

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

ERROR - The ERROR response indicates that a transfer error has occurred and the transfer has been unsuccessful.

RETRY and SPLIT - Both the RETRY and SPLIT transfer responses indicate that the transfer cannot complete immediately, but the bus master should continue to attempt the transfer.

V. CONCLUSION

I proposed a flexible arbiter based on the SM arbitration scheme for the ML-AHB bus matrix. This arbiter supports three priority policies-fixed priority, round-robin, and dynamic priority-and three approaches to data multiplexing-transfer, transaction, and desired transfer length; in other words, there are nine possible arbitration schemes. In addition, the proposed SM arbiter selects one of the nine possible arbitration schemes based on the priority-level notifications and the desired transfer length from the masters to allow the arbitration to lead to the maximum performance.

I therefore expect that it would be better to apply this SM arbitration scheme to an application- specific system because it is easy to tune the arbitration scheme according to the features of the target system. For future work, I feel that the configurations of the SM Arbitration scheme with the maximum throughput need to be found automatically during runtime.

REFERENCES

[1] Implementation of a Self-Motivated Arbitration Scheme for the Multilayer AHB Busmatrix -Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang, Member, IEEE

[2] M. Drinic, D. Kirovski, S. Megerian, and M. Potkonjak, “Latencyguided on-chip bus-network design,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 25, no. 12, pp. 2663–2673, Dec. 2006.

[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/N AMBA _ Spec.html

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

[6] R. Usselmann, “WISHBONE interconnect matrix IP core,” Open- Cores, 2002. [Online]. Available: http:// www. Openc -ores.org/ ?do=project=wb_conmax

[7] 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.

[8] D. Flynn, “AMBA: Enabling reusable on-chip designs,” IEEE Micro, vol. 17, no. 4, pp. 20–27, Jul./Aug. 1997.