Implementation of Multilayer AHB Busmatrix for ARM

E. Raja, K.V. Ramana

***Abstract*— The multi-layer AHB busmatrix (ML-AHB busmatrix) proposed by ARM is a highly efficient on chip bus that allows parallel access paths between multiple masters and slaves in a system. However, the ML-AHB busmatrix of ARM offers only transfer-based fixed-priority and round-robin arbitration schemes. In this paper, we present one way to improve the arbiter implementation of the ML-AHB busmatrix. The proposed arbiter, which is Self-motivated (SM), selects one of the nine possible arbitration schemes based upon the priority-level and the desired transfer length from the masters so that arbitration leads to the maximum performance.**

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.

***Index Terms*—ML-AHB busmatrix, Self-motivated Arbiter, fixed-priority arbitration, round-robin arbitration.**

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) busmatrix from ARM, the PLB crossbar switch from IBM, and CONMAX from Silicore. Among them, the ML-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 [5], and the good architecture of the AMBA bus for applying embedded systems with low power. The ML-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. In particular, the ML-AHB busmatrix 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 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 busmatrix is an interconnection scheme based on the AMBA AHB protocol [6], 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 busmatrix 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 busmatrix of ARM presents only transfer-based arbitration schemes, i.e., transferbased fixed-priority and round-robin arbitration schemes. This limitation on the arbitration scheme may lead to degradation of the system performance because the arbitration scheme is usually dependent on the application requirements; recent applications are likewise becoming more complex and diverse. By implementing an efficient arbitration scheme, the system performance can be tuned to better suit applications.

For a high-performance on-chip bus [1], several studies related to the arbitration scheme have been proposed, such as table-lookup-based crossbar arbitration, two-level time-division multiplexing (TDM) scheduling, token-ring mechanism, dynamic bus distribution algorithm, and LOTTERYBUS. However, these approaches employ master-side arbitration. Therefore, they can only control priority policy and also present some limitations when handling the transfer-based arbitration scheme since master-side arbitration uses a centralized arbiter. 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 busmatrix. Our SM arbitration scheme has the following advantages:

Manuscript Received September 10, 2011.

E. Raja, **Raja,** Pursuing M.Tech(ECE) in Pydah College of Engineering & Technology, Visakhapatnam, A.P., India

K.V. Ramana, Working as Assoc. Professor & Head, Department of ECE at Pydah College of Engineering & Technology, Visakhapatnam, A.P., India

1) It can adjust the processed data unit.
2) it changes the priority policies during runtime.
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. 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. EXISTING SYSTEM ARCHITECTURE

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. This limitation on the arbitration scheme may lead to degradation of the system performance because the arbitration scheme is usually dependent on the application requirements; recent applications are likewise becoming more complex and diverse. By implementing an efficient arbitration scheme, the system performance can be tuned to better suit applications.

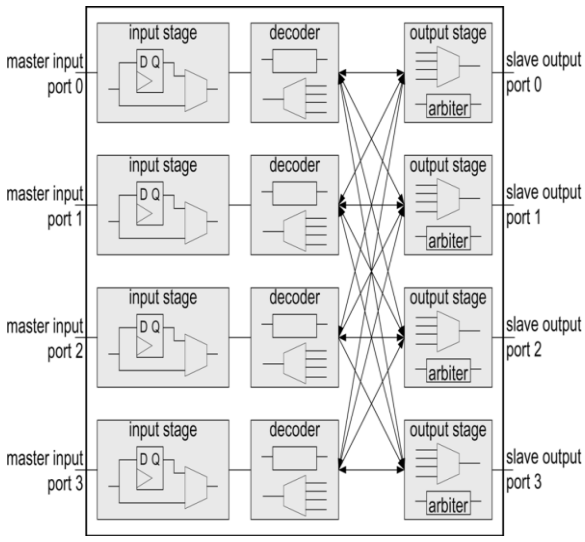

Fig. 1. Overall structure of the ML-AHB bus matrix of ARM

Disadvantages of Existing Systems

This arbitration scheme may lead to degradation of the system performance because the arbitration scheme is usually dependent on the application requirements; recent applications are likewise becoming more complex and diverse.

III. PROPOSED SYSTEM ARCHITECTURE

For a high-performance on-chip bus, several studies related to the arbitration scheme have been proposed, such as table-lookup-based crossbar arbitration , two-level time-division multiplexing (TDM) scheduling , token-ring mechanism , dynamic bus distribution algorithm , and LOTTERYBUS . However, these approaches employ master-side arbitration. Therefore, they can only control priority policy and also present some limitations when handling the transfer-based arbitration scheme since master-side arbitration uses a centralized arbiter. 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[2]. 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.

3.1 Advances of Proposed System
It can adjust the processed data unit.

1. It changes the priority policies during runtime.
2. It is easy to tune the arbitration scheme according to the characteristics of the target application.

2.2 Internal structure of Self-Motivated arbiter

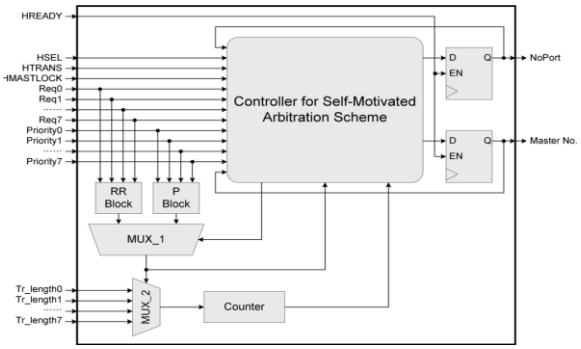

Fig. 2. Internal structure of our arbiter.

Fig. 2 shows the internal structure of our arbiter based upon the SM arbitration scheme.

In Fig. 2, the *NoPort* signal means that none of the masters must be selected and that the address and control signals to the shared slave must be driven to an inactive state, while *Master No.* indicates the currently selected master number generated by the controller for the SM arbitration scheme. In general, our arbiter consists of an RR block, a P block, two multiplexers, a counter, a controller, and two flip-flops. *MUX_1* and *MUX_2* are used to select the arbitration scheme and the desired transfer length of a master, respectively.Acounter calculates the transfer length, with two flip-flops being inserted to avoid the attempts by the critical path to arbitrate. An RR block (P block) performs the round-robin- or priority-based arbitration scheme.

IV. ARBITRATION SCHEMES FOR THE ML-AHB BUSMATRIX OF ARM

The ML-AHB [3] bus matrix 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 [4] employs 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 busmatrix of ARM furnishes only transfer-based arbitration schemes, specifically transfer-based fixed-priority and round-robin arbitration schemes.

V. SM ARBITRATION SCHEME FOR THE ML-AHB BUSMATRIX

An assumption is made that the masters can change their priority level and can issue the desired transfer length to the arbiters in order to implement a SMarbitration scheme. This assumption should be valid because the system developer generally recognizes the features of the target applications. For example, some masters in embedded systems are required to complete their job for given timing constraints, resulting in the satisfaction of system-level timing constraints. The computation time of each master is predictable, but it is not easy to foresee the data transfer time since the on-chip bus [7] is usually shared by several masters. Previous works solved this issue by minimizing the latencies of several latency-critical masters, but a side effect of these methods is that they can increase the latencies of other masters.

Unlike existing works, our scheme can keep the latency close to its given constraint by adjusting the priority level and transfer length of the masters.

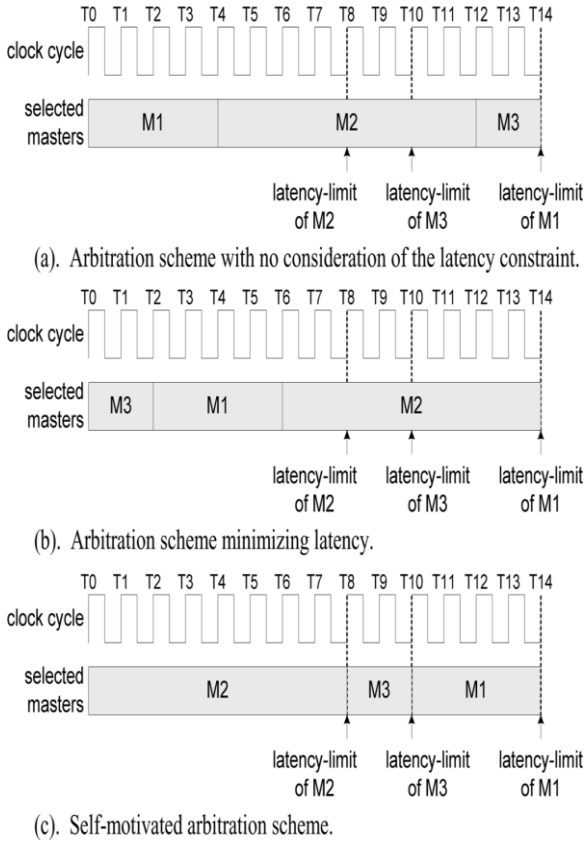

Fig. 3. Arbitration scheme examples in an embedded system.

(a) Arbitration scheme with no consideration of the latency constraint. (b) Arbitration scheme minimizing latency. (c) SM arbitration scheme.

VI. CONTROL PROCESS

A controller compares the priority levels of the requesting masters. If the masters have equal priorities, the controller selects the round-robin arbitration scheme (RR block); in other cases, it chooses the priority arbitration scheme (P block). The controller also makes the final decision on the master for the next transfer based on the transfer length of the selected master. The control process follows the following three steps.

1) If *HMASTLOCK* is asserted, the same master remains selected.
2) If *HMASTLOCK* is not asserted and the currently selected master does not exist, the following hold.
   a) If no master is requesting access, the *NoPort* signal is asserted.
   b) Otherwise, a new master for the next transfer is initially selected. If the masters have equal priorities, the round-robin arbitration scheme is selected; otherwise, the priority arbitration scheme is chosen. In addition, the counter is updated based on the transfer length of the selected master.
3) If none of the previous statements applies, the following hold.
   a) If the counter is expired, the following hold.

b)

i) If the requesting masters do not exist, the *No-Port* signal is updated based on the HSEL signal of the currently selected master. If the HSEL signal is "1," the same master remains selected, and the *NoPort* signal is deasserted. Otherwise, the *NoPort* signal is asserted.

ii) Otherwise, a master for the next transfer is selected based on the priority levels of the requesting masters. Also, the counter is updated.

b) If the counter is not expired, and the HSEL signal of the current master is "1," the same master remains selected, and the counter is decreased.

c) If the currently selected master completes a transaction before the counter is expired, the following hold.

i) If the requesting masters do not exist, the *No-Port* signal is asserted.

ii) Otherwise, a master for the next transfer is chosen based on the priority levels of the requesting masters, and the counter is updated.

The SM arbitration scheme is achieved through iteration of the aforementioned steps. Combining the priority level and the desired transfer length of the masters allows our arbiter to handle the transfer-based fixed-priority, round-robin, and dynamic-priority arbitration schemes. Moreover, our arbiter can also deal with the desired-transfer-length-based fixed-priority, round-robin, and dynamic-priority arbitration schemes.

VII. CONCLUSION

In this paper, we proposed a flexible arbiter based on the SM arbitration scheme for the ML-AHB busmatrix. Our 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. We therefore expect that it would be better to apply our 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, we feel that the configurations of the SM arbitration scheme with the maximum throughput need to be found automatically during runtime.

REFERENCES

1. 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.
2. S. Y. Hwang, K. S. Jhang, H. J. Park, Y. H. Bae, and
3. 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].
5. IBM, New York, "32-bit Processor Local Bus Architecture Specification," 2001.
6. R. Usselmann, "WISHBONE interconnect matrix IP core," Open-Cores, 2002. [Online].
7. N.-J. Kim and H.-J. Lee, "Design of AMBA wrappers for multiple clock 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.

AUTHORS PROFILE

E.Raja pursuing his M.Tech(ECE) in Pydah College of Engineering & Technology, under the guidence of K.V Ramana Assoc.Professor & Head, Dept.of ECE at Pydah College of Engineering & Technology, Visakhapatnam, A.P., India. My research Interests are VLSI Design and Digital Signal Processing.

Sri K.V. Ramana working as Assoc. Professor & Head, Department of ECE at Pydah College of Engineering & Technology, Visakhapatnam, A.P., India. His research interests are Digital Signal Processing and VLSI Design.