ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of a Special Arbitration Scheme for ML-AHB of AMBA

T.Jamadagni
M.Tech (VLSI DESIGN),
Department of ECE
Pydah Kaushik College of Engineering,
Visakhapatnam, India.

J.Mahesh Kumar
Assistant Professor & HoD
Department of ECE
Pydah Kaushik College of Engineering,
Visakhapatnam, India.

Abstract:
AMBA is one of the leading on-chip busing architecture widely used as the on-chip bus in System-on-a-chip (SoC) designs. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers, mixed priority and Round Robin transfers.

Multilayer advanced high-performance bus (ML-AHB) busmatrix employs slave-side arbitration. Slave-side arbitration is different from master-side arbitration in terms of request and grant signals. In this paper, we propose the design and implementation of a flexible arbiter for the ML-AHB busmatrix and improves the throughput by 40% compared to other arbitration schemes.

I. INTRODUCTION

Today in the era of modern technology micro-electronics play a very vital role in every aspects of life of an individual, in-creasing use for micro-electronics equipment increases the demand for manufacturing its components and its availability.The buses plays a key role in the system-on-a-chip(SoC) designs by efficient integration of of het-erogeneous system components such as CPUs, DSP and etc. To solve the bandwidth prob-lems, there have been several types of high-performance on-chip buses proposed, such as the multilayer AHB (ML-AHB) bus-matrix from ARM.[2].

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. Fig.1.The AMBA AHB is for high-performance, high clock frequency system modules.The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macrocell functions. AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated test techniques.[1].

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.[2].

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: 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.[3]. Fig .2.

In this paper our proposed SM arbiter selects one of the nine possible arbitration to allow the arbitration to lead to the maximum performance.[1].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

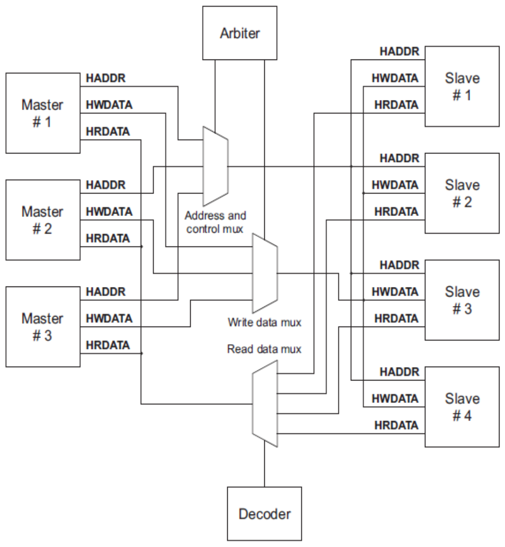

Fig.1 AMBA AHB design with 3 masters and 4 slaves and an Arbiter

II. ARBITRATION SCHEMEs FOR THE ML-AHB BUSMATRIX OF ARM

The AMBA AHB bus protocol is designed to be used with a central multiplexor interconnection scheme. Using this scheme all bus masters drive out the address and control signals indicating the transfer they wish to perform and the arbiter determines which master has its address and control signals routed to all of the slaves.[1]. The arbiter determines which input stage has to perform a transfer to the slave and decides which the highest priority is currently. A central decoder is also required to control t\he read data and response signal multiplexor, which selects the appropriate signals from the slave that is involved in the transfer.[6].

Before an AMBA AHB transfer can commence the bus master must be granted access to the bus .This process is started by the master asserting a request signal to the arbiter. Then the arbiter indicates when the master will be granted use of the bus.[4].

However, the ML-AHB busmatrix of ARM furnishes only transfer-based arbitration schemes, specif-ically transfer-based fixed-priority and round-robin arbitration schemes.[1].

III. SPECIAL ARBITRATION SCHEME FOR THE ML-AHB BUSMATRIX

The arbitration mechanism is used to ensure that only one master has access to the bus at any one time.The arbiter performs this function by observing a number of different requests to use the bus and deciding which is currently the highest priority master requesting the bus.The arbiter also receives requests from slaves that wish to complete SPLIT transfers.

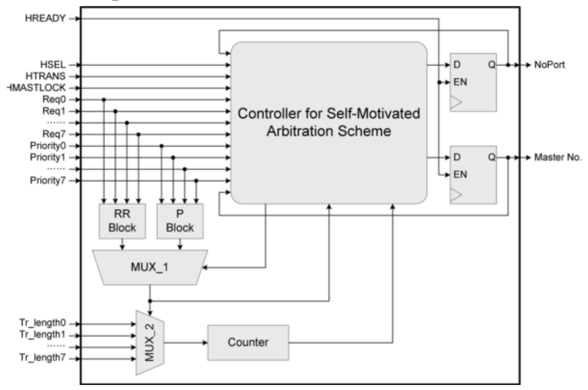

Fig. 2 Internal structure of our arbiter with SM-AS.

The fig2 shows the internal structure of our arbiter based upon the SM arbitration scheme.[1].

The role of the arbiter in an AMBA system is to control which master has access to the bus. Every bus master has a REQUEST/GRANT interface to the arbiter and the arbiter uses a prioritization scheme to decide which bus master is currently the highest priority master requesting the bus.[6].

In fig1 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. Each master also generates an HCLOCKx signal which is used to indicate that the master requires exclusive access to the bus.[1].

In fig.3. the arbiter consists of a P block, two multiplexers , an RR block, a counter, a controller and two flipflops. Both MUX_1 and MUX_2 are used to select the required arbitration scheme and selects

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

master's transfer length. An RR block (P block) performs the round-robin- or priority-based arbitration scheme. A counter is used to calculate the transfer length with two flipflops.[5]

The fig.4. shows the internal process of RR block.In this we create both up and down mask vectors(UP_Mask and Dn_Mask) based on the number of currently selected masters. Now we generate any one of the masks through bitwise between the mask vector and the requested master vector. After generating the up- and down-masked vectors, we examine each masked vector as to whether they are zero or not. If the up-masked vector is zero, the down-masked vector is inserted to the input parameter of the round-robin function; if it is not zero, the up-masked vector is the one inserted. A master for the next transfer is chosen by the round-robin function, and the current master is updated after
1 clock cycle.[1].

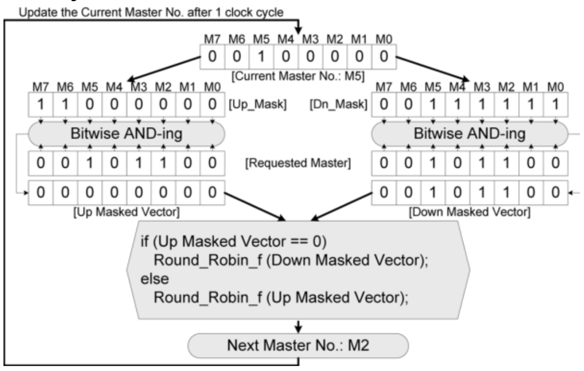

Fig. 3 Internal process of the RR block.

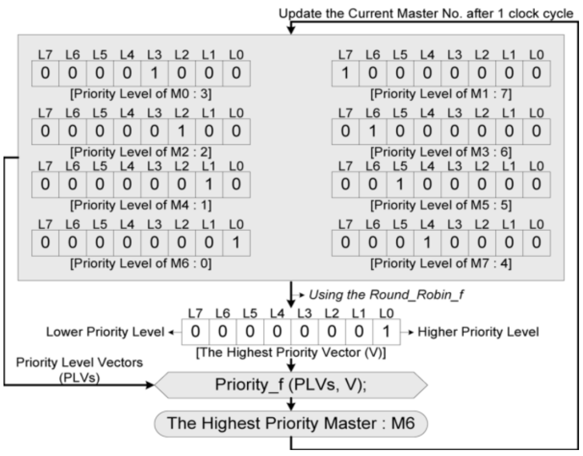
Update the Current Master No. after 1 clock cycle
L7 L6 L5 L4 L3 L2 L1 L0
0 0 0 0 1 0 0 0
[Priority Level of M0 : 3]
1 0 0 0 0 0 0 0
[Priority Level of M1 : 7]
0 0 0 0 0 1 0 0
[Priority Level of M2 : 2]
0 1 0 0 0 0 0 0
[Priority Level of M3 : 6]
0 0 0 0 0 0 1 0
[Priority Level of M4 : 1]
0 0 1 0 0 0 0 0
[Priority Level of M5 : 5]
0 0 0 0 0 0 0 1
[Priority Level of M6 : 0]
0 0 0 1 0 0 0 0
[Priority Level of M7 : 4]
Using the Round_Robin_f
L7 L6 L5 L4 L3 L2 L1 L0
Lower Priority Level
0 0 0 0 0 0 0 1
Higher Priority Level
[The Highest Priority Vector (V)]
Priority Level Vectors (PLVs)
Priority_f (PLVs, V);
The Highest Priority Master : M6

Fig. 4 Internal procedure of the P block.

Fig. 7 shows the internal procedure of the P block. First of all, we create the highest priority vector (V) . After generating the highest priority vector (V), the priority-level vectors and the highest priority vector (V) are inserted to the input parameters of the priority function. The master with the highest priority is chosen by the priority function, while the current master is updated after 1 clock cycle.[8]

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 (abbreviated as the FT, RT, and DT arbitration schemes, respectively), as well as the transaction-based fixed-priority, round-robin, and dynamic-priority arbitration schemes (abbreviated as the FR, RR, and DR arbitration schemes, respectively).[9].Moreover, our arbiter can also deal with the desired-transfer-length-based fixed-priority, round-robin, and dynamic-priority arbitration schemes (abbreviated as the FL, RL, and DL arbitration schemes, respectively).The transfer- or transaction-based arbiter switches the data transfer based upon a single transfer (burst transaction), and the desired-transfer-length-based arbiter multiplexes the data transfer based on the transfer length assigned by the masters.[7],[2].

IV. IMPLEMENTATION AND SIMULATION RESULTS

In this paper we implemented different slave-side arbitration scheme for the ML-AHB busmatrix. Each arbitration-scheme-based busmatrix was implemented with synthesizable RTL VHDL targeting XILINX FPGA (XC2VP100-6ff1704). The XILINX design tool (ISE 7.1i) was used to measure the total area. The implemented arbitration schemes which we implemented are FT, FR, RT, RR, DT, DR, and SM arbitration schemes.[10].fig,5-9.

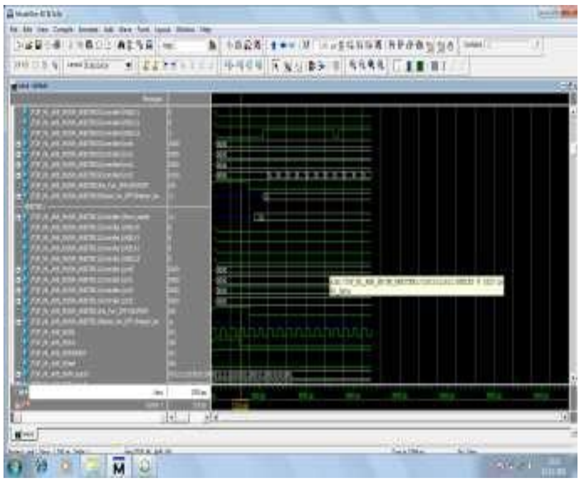
Fig.5.simulation result of the top_ml _AHB bus matrix

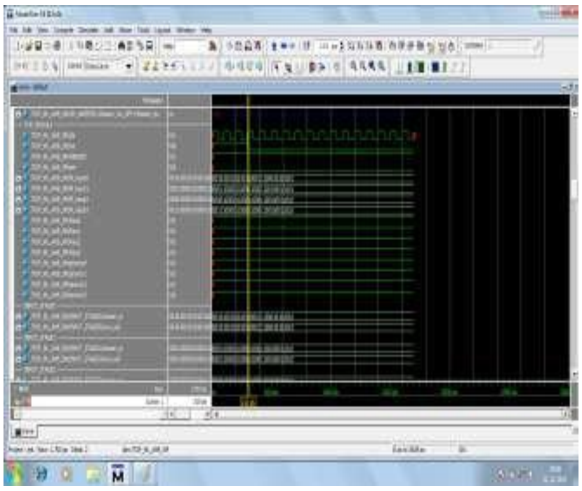
Fig.6. DR arbitration scheme

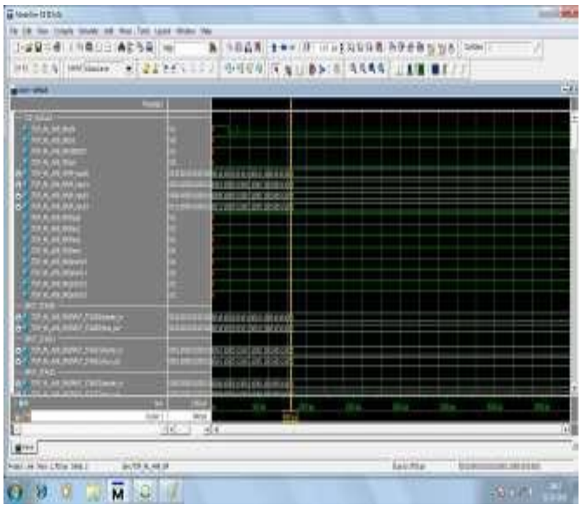
Fig.7 DT arbitration scheme

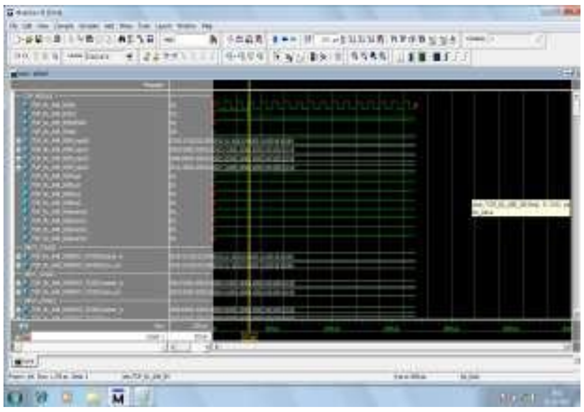
Fig.8.FL arbitration scheme

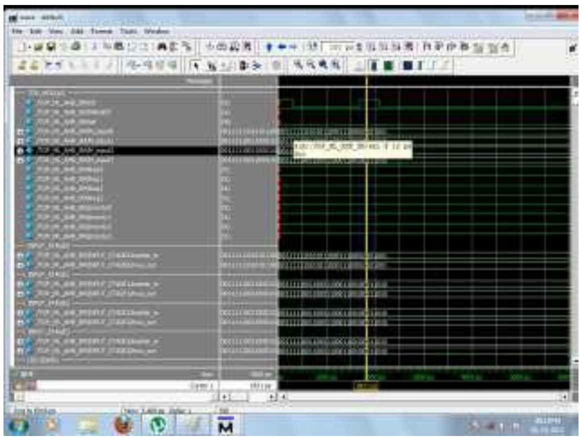
Fig.9.DL arbitration scheme

V. Conclusion

In this paper, the 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.

This design proposed ML- AHB SM arbitration schemes increases area than the other arbitration schemes in ML-AHB, but ML-AHB SM arbitration scheme gives the better performance when it selects the input stage and output stage in self motivated manner. 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

References

1.Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang, "Implementation of a Self-Motivated Arbitration Scheme for the Multilayer AHB Busmatrix".-- IEEE transactions on very large scale integration (vlsi) systems, vol. 18, no. 5, may 2010.

2. J.Mahesh Kumar, "Design And Implementation Of a Self Arbitration Scheme For Multilayer AHB Busmatrix Of ARM". IJSETR,Volume 1, Issue 4, October 2012.

3.M. Drinic, D. Kirovski, S. Megerian, and M. Potkonjak, "Latency-guided on-chip bus-network design," IEEE Trans. Comput.-Aided De-sign Integr. Circuits Syst., vol. 25, no. 12, pp. 2663–2673, Dec. 2006.

4. ARM, "AHB Example AMBA System," 2001 [Online]. Available: http://www.arm.com/products/solutions/AMBA_Spec.html

5. R. Usselmann, "WISHBONE interconnect matrix IP core," Open-Cores, 2002. [Online]. Available: http://www.opencores.org/ ?do=project=wb_conmax

6.S. Y. Hwang, H.-J. Park, and K.-S. Jhang, "Performance analysis of slave-side arbitration schemes for the multi-layer AHB busmatrix," J. KISS, Comput. Syst. Theory, vol. 34, no. 5, pp. 257–266, Jun. 2007.

7.S. Y. Hwang, H. J. Park, and K. S. Jhang, An Efficient Implementation Method of Arbiter for the ML-AHB Busmatrix. Berlin, Germany: Springer-Verlag, May 2007, vol. 4523, LNCS, pp. 229–240.

8.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.

9.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.

10.IBM, New York, "32-bit Processor Local Bus Architecture Specifica-tion," 2001.