Implementation and Performance analysis of SS Arbiter for System on chip.

[#1]G.SRILATHA **M.PRAVEEN KUMAR* *[#3]A.AJAY KUMAR*

[#1]Assistant Professor, Department of Electronics and Communication Engineering, Sri Sai Jyothi Engineering College, Gandipet, Hyderabad-75, (A.P.), India.

[#3]Assistant Professor, Department of Electronics and Communication Engineering, Sri Sai Jyothi Engineering College, Gandipet, Hyderabad-75, (A.P.), India.

**P.G. Student, M.Tech. (VLSI), Department of Electronics and Communication Engineering, Sri Sai Jyothi Engineering College, Gandipet, Hyderabad-75, (A. P.), India.*

ABSTRACT

The AMBA Advanced high performance bus (AHB) protocol design acts as an interface between different IP cores. In this work initially the investigation on the AHB is carried out and the basic commands and its working are identified and the specifications are developed for designing the AMBA-AHB using Verilog. In this paper we propose the design and implementation of a flexible SS (slave side) arbiter scheme for the AHB bus matrix based on burst operation. Basically, AHB burst operation is that a sequence of operation happens with respect to the size given and it supports only three burst sizes. The size is acting as one of the input to the master during the burst operation and after each burst operation, the master or slave will go to the IDLE stage. The AHB design contains basic blocks such as masters and slaves and the working of these blocks based on arbitration scheme. According to arbitration scheme only one master can Access the bus at any one time. Multiplexer and Decoders are used to selects the appropriate signals between master and slaves.

Keywords – SS (slave side) Arbiter, AMBA-AHB, System On chip, AHB(Advance High Performance Bus).

1. INTRODUCTION

As CMOS technology evolves and applications become more complex with increased levels of hardware and software sharing, Soc design require a system bus with high bandwidth to perform multiple operation in parallel. To solve the Bandwidth problems, there are several types of high performance on chip buses proposed, such as multi-layer AHB (ML-AHB) bus matrix from ARM, the PLB cross bar switch from IBM, and CONMAX from silicore. Among them, the ML-AHB bus matrix is widely used in Soc design due to its simplicity. The AHB will having the following features.

- burst transfers
- split transactions
- single cycle bus master handover
- single clock edge operation
- wider data bus configurations (64/128 bits).

An AMBA-based microcontroller typically consists of a high-performance system *backbone* bus, able to sustain the external memory bandwidth, on which the CPU and other *Direct Memory Access* (DMA) devices reside, plus a bridge to a narrower APB bus on which the lower bandwidth peripheral devices are located. Figure 1 shows both AHB and APB in a typical AMBA system.

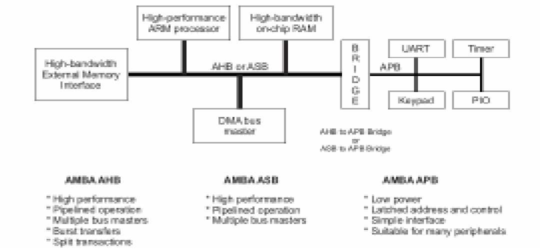


Fig1.AHB and APB Typical AMBA System

In this paper we proposed a slave side arbitration which was used for slave side. Slave side arbitration is different from master side arbitration in terms of request and grant signals, in the former merely starts a burst transaction and waits for its slave response to proceed to the next transfer. If slave will give the error response by depends on the target application we will send the new address for the same target.

The self motivated arbiter, which have the following advantages.

1) It can change the priority polices during run time.
2) It is easy to tune the arbitration scheme according to the characteristics of the target application.

2. DEMIRITS OF MASTER SIDE ARBITRATION:

For a high-performance on-chip bus, several master side arbitration schemes were proposed those are static fixed priority algorithm, TDM/Round-Robin algorithm, static Lottery Bus architecture, Dynamic lottery bus architecture. The lottery manager accumulates requests for ownership of the bus from one or more masters, each of which is (statically) assigned a number of “lottery tickets,” as shown in figure 2. The manager pseudo-randomly chooses one of the contending masters to be the winner of the lottery, favoring masters that have a larger number of tickets, and grants access to the chosen master for a certain number of bus cycles. Multiple word requests may be allowed to complete without incurring the overhead of a lottery drawing for each bus word. However, to prevent a master from monopolizing the bus, a maximum transfer size is used to limit the number of bus cycles for which the granted master can utilize the bus Also, the architecture pipelines lottery manager operations with actual data transfers, to minimize idle bus cycles.

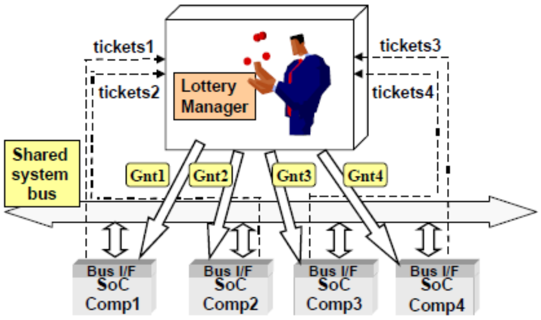


Figure 2: Lottery bus based communication architecture

3. PROPOSED SLAVE SIDE ARBITER

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 SS arbitration scheme.

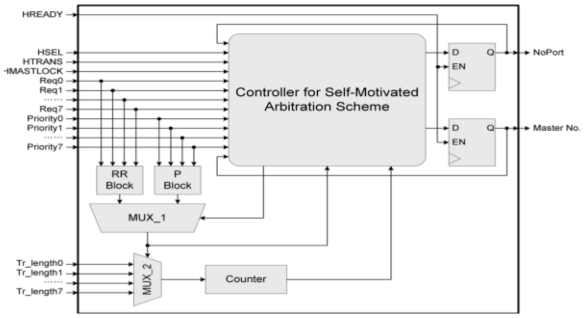


Fig. 3. Internal structure of our arbiter.

We use part of a 32-b address bus of the masters to inform the arbiters of the priority level and the desired transfer length

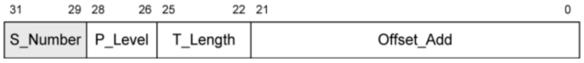


Fig. 4. Decoding information of the 32-b address bus.

The Round robin operation is given below.

```
Function Round robin(masked vector)
Variable master_no;
Variable present_state,next_state;
Variable load;
If(masked_vector≠0)then load=1;
Else load=0;
Begin
At the falling edge of clock cycle
For each load=1
If(masked_vector(i)==1)
Master_no<-i;next_state<-present_state++;
Else
Master_no<-2'bZ;next_state<-present_state;
End
Return master_no;
end
```

The operation inside the priority function is given below. Where the highest priority is to be selected.

The SS Arbitration operation is given below.

```
Function priority(priority_level,masked_vector)
Variable master_no;
Variable present_state,next_state;
If(priority_level≠0)then
Load=1;
Else
Load=0;
Begin
At falling edge of clock cycle
For each load=1
If(masked_vector(i)==1)then
next_state<-present_state++;
Else
Master_no<-2'bZ;
next_state<-present_state;
End
Return master_no;
end
```

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 *No Port* 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.

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 HSELsignal is 1,” the same master remains selected, and the *No Port* signal is reasserted. Otherwise, the *No Port* 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 priorit levels of the requesting masters, and the counter is updated.

The operation inside the SS Arbiter is given below.

```
Function controller(equ_priority, Hsel,no_port,
Master_no,cont)
At positive edge of clk
Return priority;
At negative edge of clk
If(master_no(i)==(i))
Begin
Hsel(i)<-1;
Add_out<-offset_add(i);
End
At negative edge of clk
If(c(i)_en==1)
Begin
Hsel(i)<-0;
Cont(i)<-0;
end
end
```

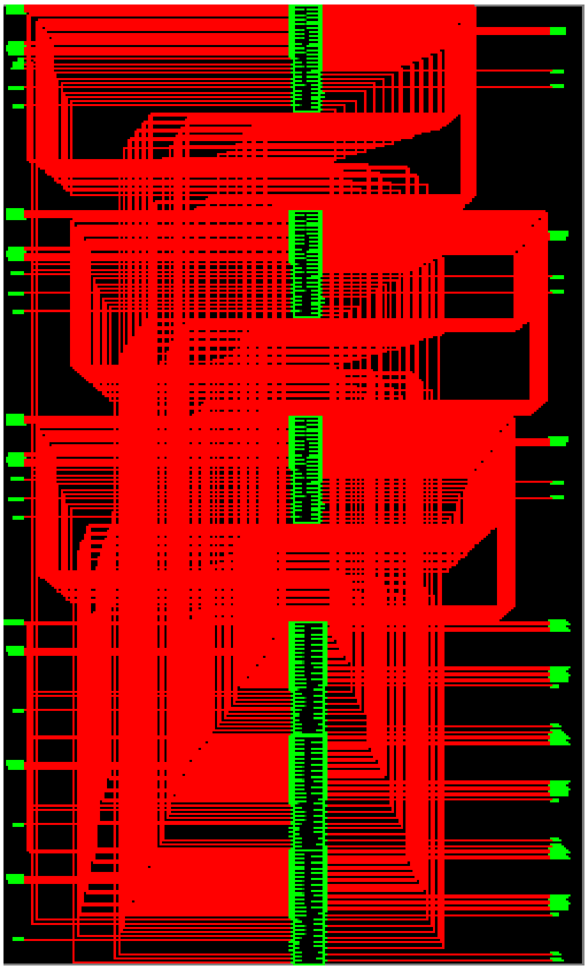
The SS arbitration scheme is achieved through iteration of the aforementioned steps. Combining the priority level and the desired transfer length of the masters allows our arbiter To handle the transfer-based fixed-priority, round-robin, and dynamic-priority arbitration schemes (abbreviated as the FT, 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). 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).

In transfer-based arbitration, the transfer length is allocated as one, which indicates single transfer. In transaction based arbitration, the transfer length is equal to HBURST signal. In addition is allocated by the demand of each master.

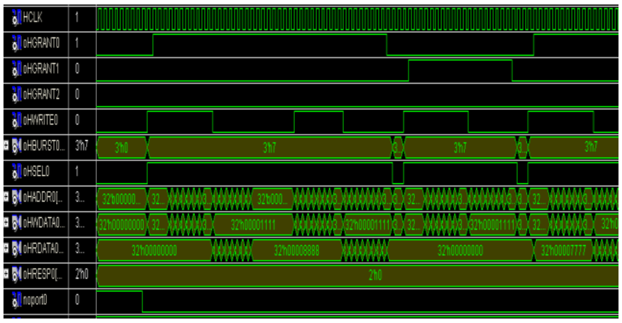
4. IMPLEMENTATION RESULTS:

We implemented the different slave side arbitration scheme for ML-AHB bus matrix. Each arbitration scheme based bus matrix was implemented with synthesizable verilog. The modelsim design tool is used for implementation of following results. In this results we observed the performance of SS Arbiter with three priority polices-Round robin, Fixed priority and Dynamic priority and three data multiplexing modes-Transfer, Transaction and Desired transfer length.

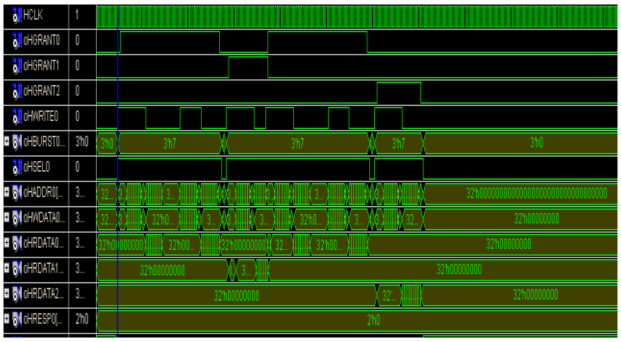
RTL-Schematic:



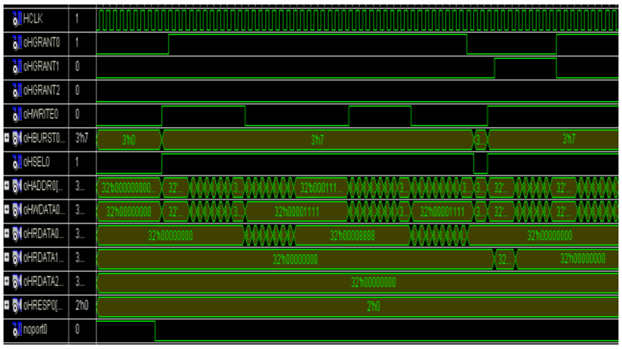
Fixed with transfer(FT)



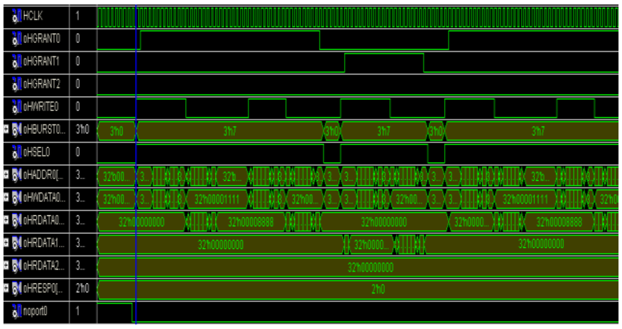
Fixed with transaction(FR)



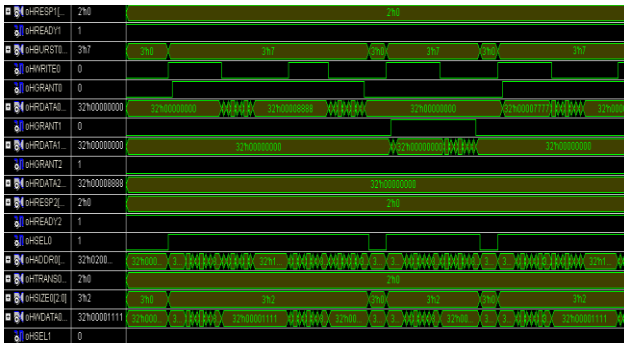
Fixed with desired transfer length(FL)



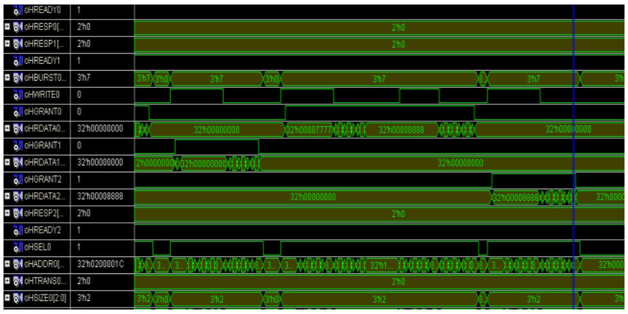
Dynamic with transfer(DT)



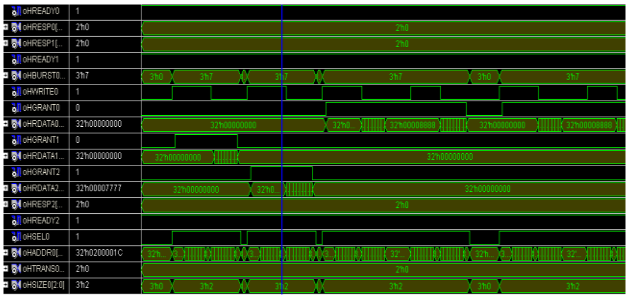
Dynamic with transaction(DR)



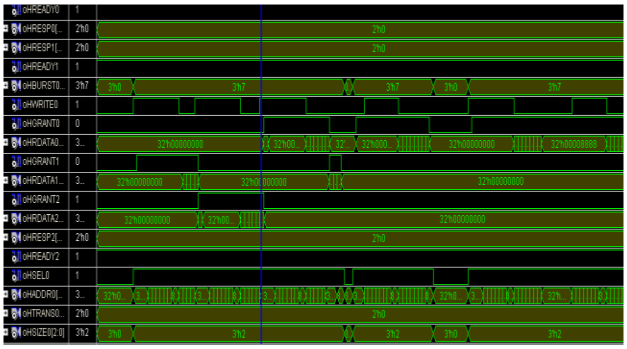
Dynamic with desired transfer length(DL)



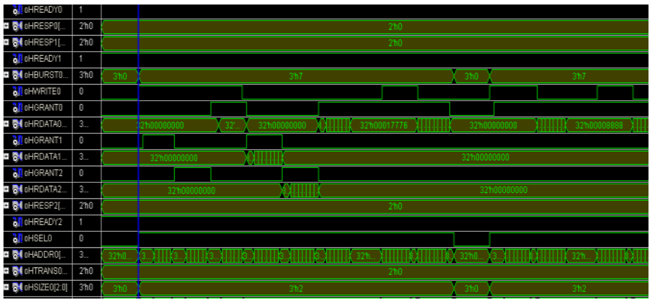
Round Robin with transfer(RT)



Round robin with transaction(RR)



Round robin with desired tranferlength(RL)



5. CONCLUSION

In this paper, we proposed a flexible arbiter which can change the arbitration scheme during the run time. It has nine arbitration schemes, out of which one will be selected at any particular time instant. Experimentally it can be proved that even though the area overhead will be little bit increased which is undesirable. Still the overall throughput increases in much higher ratio.

6. FUTURE WORK:

For future work, we feel that the configurations of the SS arbitration scheme with the maximum throughput need to be found automatically during runtime. We are likewise looking at the applicability of the proposed arbitration scheme to AMBA AXI.

7.REFERENCES

[1] M. Drinic, D. Kirov ski, S. Megerian, and M. Potkonjak, “Latency guided on-chip bus-network design,” *IEEE Trans. Computer.-Aided Design Integer. 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 H. J. Cho, “An ameliorated design method of ML-AHB busmatrix,” *ETRI J.*, vol. 28, no. 3, pp. 397–400, Jun. 2006.

[3] ARM, “AHB Example AMBA System,” 2001

[4] IBM, New York, "32-bit Processor Local Bus Architecture Specification," 2001.

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

[6] 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.

[7] D. Flynn, "AMBA: Enabling reusable on-chip designs," *IEEE Micro*, vol. 17, no. 4, pp. 20–27, Jul./Aug.1997.1http://www.arm.com/products/solutions /axi_spec.html, accessed Feb. 2008

[8] 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.

[9] S. S. Kallakuri and A. Doboli, "Customization of arbitration policies and buffer space distribution using continuous-time Markov decision processes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15,

no. 2, pp. 240–245, Feb. 2007.

[10] D. Seo and M. Thottethodi, "Table-lookup based crossbar arbitration for minimal-routed, 2D mesh and torus networks," in *Proc. Int. Conf. IPDPS*, Mar. 2007, pp. 1–10.

[11] K. Lahiri, A. Raghunathan, and S. Dey, "Performance analysis of systems with multi-channel communication architectures," in *Proc. Int. Conf. VLSI Design*, Jan. 2000, pp. 530–537.

[12] J. Turner and N. Yamanaka, "Architectural choices in large scale ATM switches," *IEICE Trans. Commun.*, vol. E-81B, no. 2, pp. 120–137, Feb. 1998.

[13] C. H. Pyoun, C. H. Lin, H. S. Kim, and J. W. Chong, "The efficient bus arbitration scheme in SoC environment," in *Proc. Int. Conf. SoC Real-Time Appl.*, Jul. 2003, pp. 311–315.

[14] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS on-chip communication architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 6, pp. 596–608, Jun. 2006.

[15] J. H. Han, M. Y. Lee, B. Younghwan, and C. Hanjin, "Application specific processor design for H.264 decoder with a configurable embedded processor," *ETRI J.*, vol. 27, no. 5, pp. 491–496, Oct. 2005.

[16] M. Jun, K. Bang, H.-J. Lee, N. Chang, and E.-Y. Chung, "Slack-based bus arbitration scheme for soft real-time constrained embedded systems," in *Proc. Int. Conf. ASP-DAC*, Jan. 2007, pp. 159–164.

[17] 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.

[18] E.-G. Jeong, J.-G. Lee, K.-S. Jhang, J.-A. Lee, and D. Har, "Asynchronous layered interface of multimedia socs for multiple outstanding transactions," *J. VLSI Signal Process. Syst.*, vol. 46, no. 2/3, pp. 133–151, Mar. 2007.

[19] S. Y. Hwang, H. J. Park, and K. S. Jhang, "An implementation and performance analysis of slave-side arbitration schemes for the ML-AHB busmatrix," in *Proc. Int. Conf. ACM Symp. Appl. Comput.*, Mar. 2007, vol. 2, pp. 1545–1551.