Design and Implementation of Power Efficient Arbiter Module for AMBA AHB Protocol

Shankaranarayana Bhat M[1], Vikrant Vijay Malode[2]

Abstract - **In shared bus architecture arbiter module plays a vital role in resolving contention between different masters to get the access for the bus. The arbiter along with resolving contention must also provide fair bandwidth allocation to each master meeting its real time requirements. In this project arbitration algorithm, taking into consideration real time requirement of masters was implemented for Advanced Microcontroller Bus Architecture (AMBA) protocol. Real time masters are given higher priority over their non-real time counterpart, and the blocking is mitigated by assigning a warning zone to the later. Clock gating technique is employed to lower dynamic power dissipation of the arbiter module. The module is designed using verilog HDL on a Xillinx 13.1 platform, and synthesis was done using Cadence RC compiler. Finally physical designing of the arbiter module was done using Cadence encounter digital implementation tool. Results obtained indicate that the proposed technique allocates fair bandwidth to all masters in the system at the same time meeting real time requirement of the masters aspiring to get access of the bus. Total power dissipation, as well as dynamic power dissipation of the module is found to be lower due to clock gating technique.**

Index Terms—**AMBA, AHB, arbiter, clock gating.**

I. INTRODUCTION

System on chip (SoC), which is also known as single chip integrated circuit, consists of several components such as programmable processors, various hardware modules to perform specific tasks, on chip memory, input output interface and communication architecture for communication among the modules. Buses are widely used for communication between these on chip modules, due to their simplicity and efficiency with which they are able to transfer the data. There have been several bus based communication architecture standards in use for on chip communication. These standards were introduced by several vendors, since 1990, in order to address emerging need of SoC based designs [1]. IBM CoreConnect is synchronous on chip bus communication protocol introduced by IBM [2]. This bus standard specifies three types of buses, namely processor local bus (PLB), on-chip peripheral bus (OPB), device control register (DCR) bus. STMicroelectronics introduced ST Bus for microcontroller based consumer applications [3]. It is closely associated to the VSIA (virtual sockets interface alliance). It's an industry interface standard. This standard makes compatibility and integration easy with IP blocks of third party. The Sonics SMART Interconnect is also on chip communication bus standard [4]. It was introduced to make component interoperability easy and also to make high performance available for wide spectrum of applications. An open source on chip communication standard is available known as Wishbone. It is a single synchronous bus specification with high speed, which connects all components introduced in an SoC design [5]. The Avalon on chip communication bus standard introduced by Altera is synchronous in nature [6].

A bus protocol is used, which facilitates the user to carry out the transactions on the bus. Bus protocol defines rules for various parameters such as defining duration, sequence, size of data , acknowledgement signals etc. for the purpose of reliable data transaction on the bus. One of the most widely used on chip communication standards today is AMBA version 2.0 [1]. The major goal of this protocol is to provide high performance bus architecture specification that is technology independent, utilizes minimum silicon area, and allows IP reuse [7]. The main focus of this work has been on Advanced High Performance (AHB) bus standard of AMBA protocol. It's a high performance bus standard which is meant to interconnect high bandwidth, high clock frequency components direct memory access controllers, high bandwidth on chip memory blocks and off chip memory interface. On a shared SoC bus, several IPs may request for the bus at the same time and contention may occur. An arbiter is an important component on a shared SoC bus, which decides who will get access to the bus in case multiple masters request for the bus at the same time. The criteria which is followed by arbiter to choose the master depends on the arbitration scheme followed by the arbiter. This arbitration scheme is application specific and is left for the discretion of the designer. There are several arbitration algorithms which are used to resolve the contention between the masters on the shared bus. The performance of the SoC depends largely on the effective communication between various components of SoC, rather than pure speed of processors. Since use of arbiter is involved during every data

Shankaranarayana Bhat M, *Associate Professor-senior, Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal University, Manipal, Karnataka, India*

Vikrant Vijay Malode*, M.Tech(Microelectronics), Manipal Institute Of Technology, Manipal University, Manipal, Karnataka, India*

ISSN: 2278 – 1323

transfer, they are considered as a vital element of on chip communication architecture and emphasize the need for a careful design.

Power consumption has become a matter of concern in designing of portable devices consisting of integrated circuits [8]. Switching activities in the circuit leads to dynamic power dissipation. Higher frequency of operation is the major source of switching activity which results in increased dynamic power dissipation. Clock signal which has to carry lot of load, as it has to reach all sequential logic on the chip, becomes major contributor to dynamic power dissipation. If microprocessor of high performance is taken under consideration then it has come to the noticed that clock distribution dissipates 40% of the total power [9]. Clock gating is one of the accepted design technique to optimize dynamic power which can be implemented at various design stages, namely – system level, gate level, or architectural level. The amount of effect clock gating has on overall power dissipation depends on fact that, at which of the design stage it is applied [10].

Today there are several arbitration schemes available for scheduling request for bus based architecture. The main motive of these techniques is to allocate sufficient bandwidth, avoid starvation, assigning higher priority for the critical data and ensure ease of access for all masters. Static priority algorithm is probably the most simplest and popular arbitration algorithm. In this algorithm each master is assigned fixed priority. When multiple masters request for the bus, the master with highest priority always gets access to the bus. It ensures high performance by assigning higher priority to crucial data transfers such as between processors and memory. [1]. But the main problem associated with this scheme is the possibility of starvation of lower priority masters. If there are frequent request from higher priority masters then there is possibility that lower priority masters may not get chance to access the bus. Also, this scheme cannot address the real time requirement of masters. Another popular arbitration scheme is round robin algorithm which provides solution to starvation problem [11]. In this technique a token (bit) is being rotated in a cyclic manner from one master to another. If a particular master is having token i.e. the token bit is set and also the request bit for the particular master is set, then bus is granted to that particular master. This scheme overcomes the drawback of static priority algorithm i.e. starvation. But the problem with this technique is critical data transfer may have to suffer from latency problem as the master have to wait for their turn to come. Also real time request cannot be served by this technique. K. Lahiri et.al proposed LOTTERYBUS which is also known as probabilistic arbitration algorithm [12]. In this technique each master is assigned 'lottery ticket'. There is a module called as 'lottery manager' which accepts request from all the masters, which desires to own the bus. A pseudo random number is generated by lottery manager. The master whose ticket matches with this number is granted the access of the bus. To meet the real time requirement of all masters, proper number of tickets must be assigned according to their criticality and bandwidth requirement. It is very difficult to assign tickets to masters with diverse real time and bandwidth requirements. If a master is having a hard real time requirement but requires a small fraction of bandwidth probabilistic lottery approach proves to be inefficient. Earliest deadline first, also known as least time to deadline algorithm is another simple algorithm to schedule the requests of real time masters [13]. It takes into consideration the deadlines associated with real time masters and accordingly schedules the requests. Though this technique is simple and efficient in scheduling real time requests, it fails to perform non real time requests are pending along with real time requests. This leads to huge latency issue for non real time master.

The main objective of this work is to implement an arbiter module which takes into consideration bandwidth requirement along with meeting of real time requirement of the masters. Many arbitration algorithms available today fail to meet both these basic requirements simultaneously. The arbitration algorithm should also see to it that none of the masters suffers by starvation. The algorithm should ensure that any master operating in real time scenario has to meet its deadline. As AMBA AHB is the back bone of high frequency and high performance application, dynamic power dissipation is also one more parameter of importance. In this work we also address reduction of dynamic power dissipation by employing low power techniques.

II. METHODOLOGY

This paper intends to provide solution to contention between real time and non real time master, if they request for the bus simultaneously. The basic idea here is, to set a warning zone for all real time masters [14]. The master which enters the warning zone will be considered having highest priority and will be granted with the bus. As soon as any real time master issues a request, its deadline and warning zone length will be calculated. The masters with real time constraints which has requested and is waiting for its turn to get access to bus, has its deadline counter which is continuously decrementing. If it's time to deadline goes below warning zone length, it will be allotted highest priority and will be granted access to the bus. In a scenario, where, more than one masters have already entered their warning zone, then, the master with lowest time to deadline among them will be given highest priority so that it will be given the grant of bus. If, on the other hand, a non real time master is competing with a real time master for the bus, then, non real time master will be granted with the bus if and only if, real time master has not entered its warning zone. That means, in any case, non real time master will not get higher priority over real time master, if the latter is when a in its warning zone. Illustration of proposed technique is shown in figure 2.1. The deadline is summation of arrival time, execution time and slack. Slack is calculated as flexibility factor times execution time [15].

Deadline = Arrival time + Execution time + slack (1)

Slack = flexibility factor x Execution Time (2)

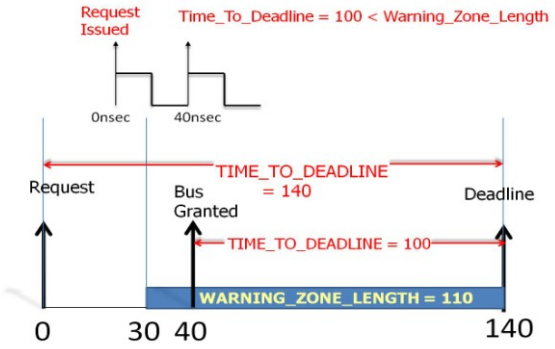

Fig 2.1: Illustration of proposed approach.

This slack is introduced to make timing constraints less stringent. It needs to accommodate bus handover latencies, opcode fetch or data fetch latencies and other latencies such as wait state introduced during transaction, to respond to RETRY/ERROR responses given by slave [15]. As the flexibility factor is left to the designer to decide, we assume flexibility factor of 0.75 has been chosen for the proposed arbiter. Warning zone length is the average of execution time and the deadline.

$$\text{Warning zone length} = \frac{\text{Execution time} + \text{Deadline}}{2} \quad (3)$$

Clock gating at architectural level of abstraction is very effective in reducing dissipation of dynamic power. At this stage, design is synchronized using clock and the lower stages of design supplies information on power accurately. The designer only has to decide when and to which logic block the clock must be gated. All flip flops with common enable input are identified and such common enable line is used to control the clock gating logic. On enabling the clock gating, the gated blocks will not toggle and hence no dissipation of dynamic power. This will in turn reduce the total power consumption.

A. Simulation Model

We assume the proposed arbiter follows non idling and non pre-emptive policy while arbitration. In idling scenario, when there is no request form any master, bus is granted to a default master which perform idle transactions. In non pre-emptive case, if bus is allotted to a master, the master will not be interrupted until and unless the transaction is complete. Width of data bus assumed is 32 bits wide and clock frequency assumed is 25MHz, giving a clock period of 40nsec for simulation purpose. The real time masters in this simulation model are assumed to perform firm real time task. They do not perform hard real time task. If the real time master performing hard real time task fails to meet the deadline, the system will fail. But such is not the case with real time masters performing firm real time task. The results which are obtained after the deadline are discarded and the system will continue to function normally. Three types of masters were discussed in [16], which is considered here for simulation purpose and are described below:

1. Dependent non real time type master.

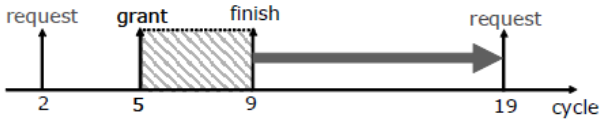

Fig 2.2: D type master with R_{cycle} = 10 [17].

These are non real time type of masters and they don't have any deadline requirement. There is predetermined number of cycles between the finish time of previous request and issue time of next request. In figure 2.2 R_{cycle} is set as 10 cycles.

2. Dependent real time type master.

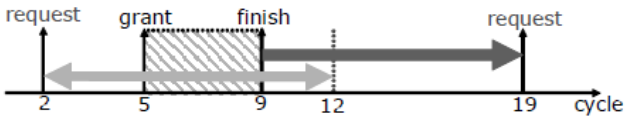

Fig 2.3: D_R type master with R_{cycle} = 10 [17].

Dependent real type of masters has a specified deadline but they are same as real time masters of dependent type. They are known as dependent because the issue time of next request depend on the finish time of previous request. In figure 2.3 this dependency is specified in terms of R_{cycle} which is considered as 10.

3. Non dependent real time master.

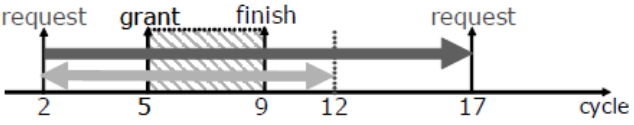

Fig 2.4: ND_R type master with R_{cycle} = 15 [17].

Non dependent real time type of masters has a specified deadline. They are known as non dependent because the issue time of next request do not depend on the finish time of previous request. In figure 2.4, we can see that the time interval between two requests is specified in terms of R_{cycle} which is considered as 10.

III. RESULTS AND DISCUSSION

Code for the project was written in verilog HDL and was simulated using Xillinx's, ISim design simulator, and version 13.1. Bandwidth for dependent real time and non dependent real time master was calculated over a particular amount of time period. The result of which is tabulated in table 3.1. As non pre-emptive policy is followed some of the request were failed to address as the bus was busy transferring the data of other master. Non real time master

ISSN: 2278 – 1323

have their entire request addressed i.e. they get the total required bandwidth.

Table 3.1: Bandwidth results for real time masters.

Type of Master	Required Bandwidth	Allotted Bandwidth	% of Request Failed to Address
Dependent real time	40.85%	35.65%	33.93%
Non dependent real time	54.35%	49.4%	24.16%

Synthesis of the verilog HDL code has been done using Cadence RTL compiler, version 9.1.201. 180 nm technology library, of standard cells is used. Figure 3.1 show the schematic view of RTL after synthesis takes place. The tool generates area, power and timing report for the design. After the synthesis is done the tool generates .sdc file which forms input to encounter digital implementation tool for physical design.

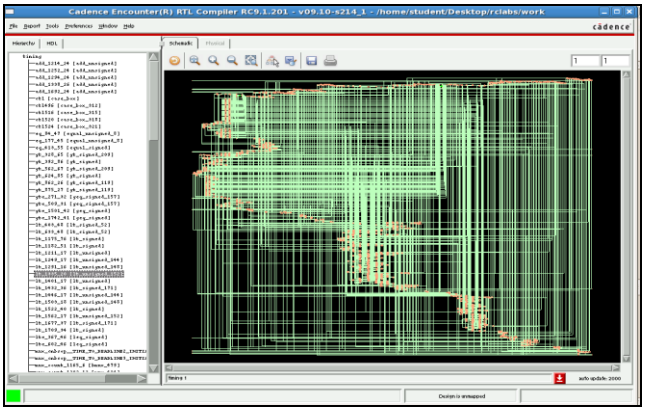

Fig 3.1: Schematic view of RTL for proposed arbiter.

Table 3.2: Synthesis results.

Result Parameters	Without Clock Gating	With Clock Gating	% of Change After Clock Gating
Number of Cells	994	1009	+1.5
Total Area (sq. micron)	5299	5396	+1.8
Dynamic Power Dissipation (nW)	74698.48	67034.42	-10.26
Total Power Dissipation (nW)	92154.08	83483.38	-9.41
Timing slack (ps)	301	302	-

From table 3.2 it can be noticed that number of cells as well as the area is increased after clock gating. There is a reduction in dynamic power dissipation due to the insertion of clock gating to the design. Physical design of the arbiter module was done using encounter digital implementation tool, version 9.12 of Cadance. The figure 3.2 below shows die after placement of standard cells. Standard cells from 180nm technology library are used.

After placement of standard cells has been done, timing analysis is performed to check set up and hold time violations. Figure 3.3 shows the chip after clock tree synthesis and final routing has been done. At each stage the design is optimized to avoid any violating path in the design. Table 3.3 gives information about setup timing analysis. This data is obtained after optimizing the design at post routing stage. Similarly, table 3.4 shows the results of timing analysis for optimized design after routing has been done. From the timing analysis results, it is clearly seen that there are no setup and hold time violations.

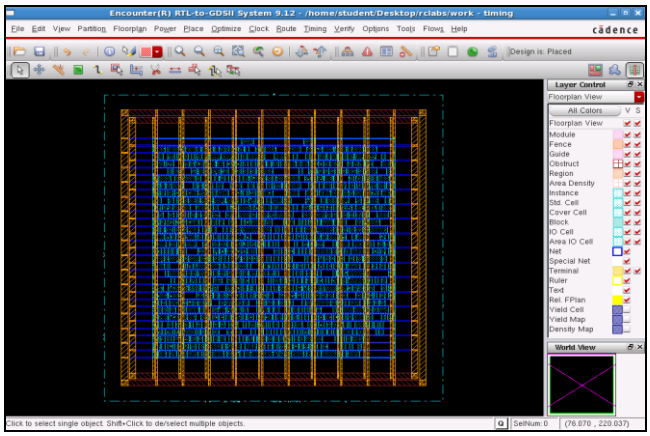

Fig. 3.2: Snapshot showing placement of standard cells.

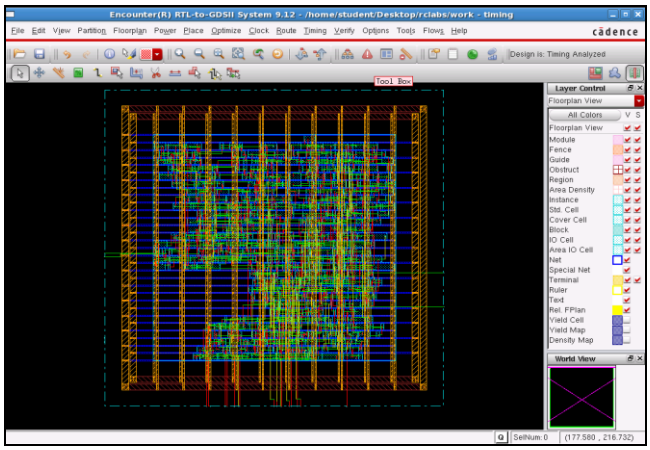

Fig. 3.3: Snapshot of the chip after final routing.

Table 3.3: Setup analysis after final routing.

Setup Mode	All	Reg to Reg	In to Reg	Reg to Out	In to Out	Clock gated
WNS (ns)	0.054	0.054	0.000	0.036	0.037	0.049
TNS (ns)	0.027	0.015	0.000	0.000	0.000	0.012
Violating paths	0	0	0	0	0	0
All paths	47	14	15	3	2	13

Table 3.4: Hold analysis after final routing.

Hold Mode	All	Reg to Reg	In to Reg	Reg to Out	In to Out	Clock gated
WNS (ns)	0.084	0.025	NA	NA	NA	0.084
TNS (ns)	0.148	0.000	NA	NA	NA	0.148
Violating paths	0	0	NA	NA	NA	0
All paths	23	14	NA	NA	NA	9

IV. CONCLUSION

We have implemented successfully arbiter module for AMBA AHB protocol. From the results obtained it is clear that the proposed arbiter module is capable of allocating fair bandwidth to all the masters along with meeting their real time requirements. The module is also power efficient, as is evident from the results of power analysis obtained after synthesis. It is found that the total power has been reduced by 10.26% and dynamic power dissipation has been reduced by 9.41% after clock gating technique at architectural level has been applied. Though this arbiter is capable of meeting the real time requirements of masters along with fair bandwidth allocation, there are scopes for further improvements. Percentage of requests which are not addressed needs to be brought down. The difference between real time master's required bandwidth and allotted bandwidth needs to be reduced. Efforts need to be taken to achieve higher reduction of power by keeping check on area and delay overheads.

REFERENCES

[1] S. Pasricha and N. Dutt, On Chip Communication Architecture, Morgan Kaufmann.
[2] IBM CoreConnect Specification, http://www.ibm.com.
[3] "STBus communication system: concepts and definitions," Reference guide, STMicroelectronics, May 2003.
[4] Sonics SMART Interconnect, http://www.sonicsinc.com.
[5] Wishbone Specification, http://www.opencores.org/wishbone.
[6] Altera Avalon Interface Specification, April 2006, http://www.altera.com.
[7] AMBA specification Rev 2.0, http://www.infocenter.arm.com.
[8] C. Piguet, Low-Power CMOS Circuits - Technology, Logic Design and CAD Tools, Taylor & Francis Group.
[9] G. Yeap, Practical Low Power Digital VLSI Design, Kluwer Academic Publishers.
[10] "Utilizing Clock Gating Efficiency to Reduce Power", http://www.eetimes.com.
[11] I. Singh and D. Gupta, "A priority based round robin CPU scheduling algorithm for real time systems," International Journal of Innovations in Engineering and Technology, vol. 1, Oct 2012.
[12] K. Lahiri, A. Raghunathan, G. Lakshminarayana, "The LOTTERYBUS on-chip communication architecture," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 596-608, 2006.
[13] "Introduction to Real-Time Systems", http://www.nptel.ac.in.
[14] L. Zhang, R. Wu, Y. Yang, F. Lu, "An adaptive dynamic and real-time guaranteed arbitration algorithm for SoC bus communication," Journal of Computational Information Systems, vol. 9, pp. 5693–5700, 2013.
[15] B. Kao and H. Garcia-Molina, "Deadline assignment in a distributed soft real-time system," IEEE Transaction on Parallel and Distributed System, vol. 8, pp. 1268-1274, December 1997.
[16] M. N Akhtar and O. Sidek, "An intelligent adaptive arbiter for maximum CPU utilization, fair bandwidth allocation and low latency", IETE Journal of Research, vol. 59, pp. 48-54, Jan-Feb 2013.
[17] C. H. Chen, G. W Lee, J. D. Huang, and J. Y. Jou, "A real-time and bandwidth guaranteed arbitration algorithm for SoC bus communication," In Proceedings of the Asia South Pacific Design Automation Conference, pp. 600–605, 2006

Shankaranarayana Bhat M obtained his M.Tech from IISc, Bangalore and is currently working as Associate professor-Senior in the department of Electronics and communication Engineering, M.I.T Manipal. He is also heading the Education Technology Cell of M.I.T. Manipal. In addition to contributing as resource person for technical workshops and conferences, he is also a recognized faculty trainer and successfully conducted faculty training in various Engineering colleges in India. He has presented technical papers, chaired many national and International conferences and reviewed technical papers for conferences and journals. His interests include Low Power VLSI Design and Processor architecture in addition to Engineering Education and Soft skills.

Vikrant Vijay Malode has completed his M.Tech in Microelectronics from Manipal Institute of Technology, Manipal University, Manipal in May 2014. His areas of interest include Digital VLSI design, Physical design and verification.

ISSN: 2278 – 1323