Performance Verification of Multi-Master AHB Bus System

Swetha Ittige Narendrappa, Sindhuja Yadiki RadhaSwamy, and Lili He, *Member, IAENG*

***Abstract*—The AMBA (Advanced Microcontroller Bus Architecture) AHB (Advanced High Performance Bus) master may be a high-performance memory controller or a processor or a DSP, whereas APB (Advanced Peripheral Bus) connects peripherals. The project involves Multi master AHB's performance verification- latency and bandwidth, using the test bench written in System Verilog and UVM (Universal Verification Methodology). A UVM test bench contains Verification Components which are reusable in verification environment. The verification analysis is done on multiple designs of CAN transmitter and multi master AHB with Priority and Round Robin Arbitration mechanism. The result obtained showed the application specific arbitration mechanisms to be implemented with AHB according to the Performance requirement.**

***Index Terms*—AHB, UVM, Performance – Latency, Bandwidth.**

I. Introduction

A SOC (System on Chip) may include different IP's(Intellectual Property) like memory, I/O peripherals and processors with different functionalities. These may vary in their speed and interconnection of these IP's will be important and it is done using SOC bus. AMBA was given by ARM which provides different kinds of buses to be used in microcontrollers, SOC's and ASIC's. There are mainly two kinds of buses, one to connect high performance devices and another to interact with low bandwidth peripherals. An AHB (Advanced High Performance), was designed in this project which is mainly used in industries to connect high performance memory controllers and processors. AMBA came up with APB (Advanced Peripheral Bus) to connect peripherals and AHB to connect with high performance devices. This report explains how the performance is verified using Universal Verification Methodology of multi master AHB bus which was designed using System Verilog in our curriculum. The reason for selecting AHB over APB is that APB has massive memory I/O accesses, latched address control, no pipeline it is proposed for the use connecting simple peripherals, used for low bandwidth control accesses and low complexity signal [1].

AHB is mainly used to design efficient design which require high performing bus and which run at higher clock frequencies. AHB as an interconnect plays an important role in high bandwidth devices like memory interfaces, off chip memories and the processors. AHB is the most distinct bus of AMBA which targets for lower design complexity, higher frequency and bandwidth. The other advantage of AHB are high performance, pipelined operation, multiple bus master, burst transfer and spilt transactions. It is a shared bus protocol for multi master and slaves and higher bandwidth is achieved through burst data transfers. To begin the transfer on AHB, the master should request for the bus, and get a grant to use the shared resource. The grant is given by the arbiter and it depends on the arbiter whether to give a grant or not to a particular master. This will be done by designing different kind of application specific arbiters.

The verification environment is going to be built for a single master and then the objects can be created for 7 other masters with same monitors and scoreboards. The advantage of UVM is to develop the verification components individually and to reuse them in different configurations. The main goal is to verify the AHB bus latency and bandwidth which plays a major role in defining the efficiency of AHB bus. The obtained simulation results will be run through a python script to choose which design is better with what kind of arbitrator. If there is any trade off to be made to keep the latency low or bandwidth to be high, the results will be analyzed and a middle ground can be chosen to have a win- win situation or to go for an ideal bus.

II. Motivation and Application

The motivation behind doing this is the curiosity developed about the AHB and the use of AHB in SOC's as an interconnect bus. Though there are other high-speed buses like AXI, the cost and bandwidth of AHB makes it suitable for on chip communication bus between high performance devices. The AHB designed in this project is checked for arbitration latency with two different arbitration mechanisms and verified for the best suitable arbitrator for specific applications.

AHB being a on chip bus standard is a full duplex parallel communication bus intended to be an internal bus with multiple masters and slaves inside ASIC/FPGA [Application Specific Integrated Circuits / Field Programmable Gate Arrays]. It is mainly used as an interconnect between processors, internal and external memory controllers and other high bandwidth devices. The newer AHB's are aimed towards synthesizable, DFT friendly ASIC/SOC [System on Chip] designs [2].

III. Advanced High-Performance Bus

AMBA's new generation bus is AHB which can meet the necessity required for high-performance synthesizable

Manuscript received March 5, 2018; revised April 18, 2018.
The authors are with San Jose State University, Department of Electrical Engineering, San Jose, CA95192, USA. (Corresponding author, phone: 408-924-4073; fax: 408-924-3925; e-mail: lili.he@sjsu.edu).

designs, automated techniques and also sustains the external bus interface data rates. It's a new level bus over the ASB and APB, which implements the features necessity for high clock frequency and high-performance system including split transactions, single – edge clock operation, burst transfers, bus configuration with wider data, Bus master handover in a single cycle, Pipelined operation and non-tristate implementation [3]. One or more bus masters are present in AHB design (AHB supports up to sixteen masters,) in certain conditions in a system the processor can be used as AHB master. However, DSP or DMA acts 4 as the bus masters and SRAM, APB Bridge, ROM and other internal memories act as the slave to AHB. The bus has 4 major components mainly slave, arbiter, master and decoder [4].

A. Operation

With the central multiplexer interconnect scheme the bus protocol is designed, the read and the write operations are initiated from bus master by providing the control and an address information during the first cycle. Arbiter decides which bus master information is to be transmitted to the slave since, Bus can be accessed by only one particular master at a time. Arbitrator's important role is resource sharing, depending on various arbitration mechanisms like Round Robin and Priority etc., it takes the decision of grating the bus to a particular master. Central decoder controls the response signal from the multiplexer and read data, the slave responds the failure or the success to the active master regarding the data transfer. The control and address signals carry the information regarding the direction, transfer width, burst transfer and address. There are two phases of every transfer: Control and Address cycle phase and data phase.

The slave samples the control and address signals during address phase since, this phase cannot be extended. In data phase by using HREADY signal it can be extended, when HREADY is low wait states are inserted between the transfers and indicates the transfer completion when HREADY is HIGH. Master uses Burst mode to complete all the data transfer before the arbiter gives bus access to another master [4].

B. Arbiter

Arbitration technique is to ensure that only one master can access the bus at a time. In our project Round robin and priority arbitration technique are used to grant the bus to master. This technique is performed by monitoring the request signals to lock the bus and deciding the master with high priority requesting for the bus. The slave sends the request signals to arbiter for completing the SPLIT transfers. The arbitration process is not necessary for the slaves which are not able to execute the SPLIT transfers whereas, when ownership of bus changes burst transfer may be completed. The arbitration signals are HBUSREQx, HGRANTx, HTRANS, HCLKx and HRESETx, HLOCKx, HSPLIT, HMASTER, HMASTLOCK. To request for access the bus, the bus master uses the HBUSREQx signal and arbiter output HGRANTx which specifies that a particular master is highest priority master considering the split and locked transfers. HLOCKx specifies to arbiter that the bus access should not be granted to any other master once the locked transfers have started since a number of indivisible transfers are to be performed. The address bus ownership is obtained when HGRANTx and HREADY is high at the positive edge of HCLK. Through 4 bits, HMASTER signal from arbiter specifies which master has currently access to the bus. When HMASTLOCK signal is asserted the arbiter specifies that locked sequence is been transferring, with having the same timing for control and address signals. 16-bit HSPLIT is used to indicate which bus master has completed the split transaction [5].

In priority arbitration, all the masters have given a priority range, highest priority master from all the masters gains the bus access. During increasing in communication traffic situation, this type of arbitration leads to low priority devices in a starvation. Because, during the heavy traffic high priority devices keep requesting for bus and low priority requests are not considered during this time which leads to not getting bus access. To overcome the starvation problem round robin arbitration used [6].

Round Robin arbitration is a fair and simple mechanism in which no device will get a bus access indefinitely. Master gets the bus access in a particular manner. When a particular master turn ends either by unused because of data transfer completion or by limited time period access is given to next master. The disadvantage of this mechanism is that it checks all the masters even if there are no pending requests. Hence, reducing the system performance from the distribution latency. Giving all the masters equal bus share is not a good approach as more frequent bus request masters are treated as idle masters. To overcome this disadvantages other bus arbitration mechanism can be used like token passing or bandwidth arbitration [7].

IV. Design and Verification steps

To begin with, first the devices to be placed on AHB, the CAN (Control Area Network) is a multi-master serial bus standard which are extensively used in automotive applications, is designed with CRC, bit stuffing and bit timing basically as a low-level protocol without much security features. At message level error checking is enforced with CRC, 16 bits which contains the preceding data application checksum. CAN basically has 4 kind of message frames/types, the data frame, remote frame, error frame and the overload frame. In this project, the CAN handled only data frame and remote frame [8].

System Verilog Interfaces were used to connect different modules, as a new port type, which allows to group signals as a single port. Interfaces help in reducing the error caused during interconnection between modules. It also helped in adding or deleting the signals as it can be made as a separate file, which gives an added advantage for reusability.

AMBA AHB bus intended to be a high-performance bus supporting multiple masters was designed as a part of course project. It is pipelined bus where in first cycle address and control signals are sampled and in next clock cycle the data is sampled. The AHB bus system contains a AHB master, a slave, an arbiter and AHB decoder. The designed AHB used a central multiplexor to interconnect the bus masters, arbitration signals and the slave. The bus masters send out the address and control signals they wish to send it to a particular slave and its arbiter's which decides which master

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

gets the bus. There is a central decoder designed to handle the slave reads and responses.

The AHB acted as a master with one CAN on it as a device, along with a test bench master. Then an arbitrator was designed and 8 devices were placed with 8 AHB masters. The parallel communication between multiple masters and slaves is not allowed in a single shared bus at the same time. To resolve the conflict between multiple masters an arbitrator was designed which gives grant at a time only to a single master to access the bus [10]. The arbitrator design was a challenge, and to begin with, for the project, priority arbitrator was designed. Each bus master was assigned a fixed priority and the grant was given to the master with highest priority after it requests. It is common to use priority arbiter if there are few masters. If the bus is kept busy because of a high priority master, the least priority master may wait forever in time to get the bus access and to get a grant from the arbitrator.

To overcome the above shortcoming and to check the performance, round robin the fair arbitration mechanism was designed. Every master on the bus is given the bus access by turn. If the first master is active and requesting for the bus, it gets the grant or else if second master is requesting it gets the grant. The number of bus masters limits the maximum wait time for the bus masters.

One can prefer the priority or round robin or even have the option for bandwidth arbitrator depending on the application. After designing the arbitrator, the 8 masters are attached to AHB bus master which can communicate with 8 slave devices depending on the device address. The UVM environment was created for a single master and each environment was replicated for the 7 similar environments except with change of device addresses

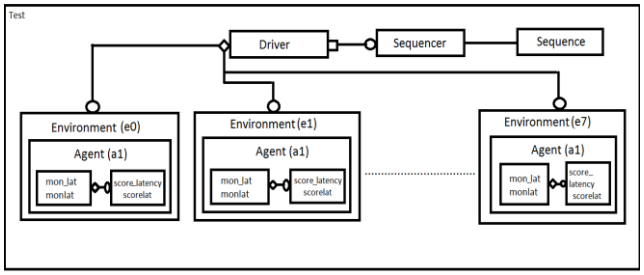

Fig. 1 UVM Verification Environment for the Project

The UVM verification was not built like in a traditional way rather as it can be shown through diagram the Test contains a Driver, Sequencer, Sequence along with 8 different environments. Even the Agent, does not include driver, rather it contains different monitors and scoreboards. And the agent object is instantiated in environment and through exports the different objects speak with each other.

Along with testing the functionality by creating a test environment, the main intention was to measure the AHB bus performance, the bus latency and bandwidth. To measure the latency and bandwidth separate monitors and scoreboards were created. The latency involves many added latencies, the main one being the arbitration latency which can vary the performance of AHB bus. The bandwidth for the designed bus is low as the data consumed by CAN transmitter is less. If the devices attached take lot more data, then the curve would have been more interesting.

The bandwidth and latency results got from the simulation result is by time stamping the request and grant signals in arbitrator and taking the difference of time so that, latency can be obtained. Python script is used to analyze the simulation results, to obtain a graph or plot latency (in clock cycles) versus time and bandwidth of both priority and round robin arbitration results in one graph plotted as bandwidth versus time.

The result obtained by plotting helped to analyze about the latency was more in priority arbitration method but bandwidth was better compared to low latency and low bandwidth in round robin. There will be a tradeoff between latency and bandwidth, if one goes with priority. The low latency gives an indication regarding the bus being kept busy because of other priority masters and it is a good measure of bus not being idle. So, one can definitely go with priority arbitration algorithm though the latency is little more compared to round robin arbitration.

V. Results

To start off, CAN (control Area Network) was designed and synthesized and placed as a device on AHB master. To verify the AHB performance as a multi master, CAN was instantiated 8 times along with 9th master, a test bench. Later, the latency and bandwidth was verified by creating a UVM environment having driver, sequencer, monitors and scoreboards.

The system level bus performance mainly narrows down to the bus latency and bandwidth. Latency measures delay in data transmission across the bus, for a single or multiple transfers to finish. Bandwidth specifies the physical limit on bus for transfer speed usually in Bits/sec. The bus can be made wider to increase the bandwidth but limited by area and also the higher frequency limits the power consumption.

There will be bus protocol latency involved which cannot be avoided, one to send out the address and one for the data, which is pipelined addressing mechanism where address and data phase can overlap if master does a burst transfer.

Second latency measure is the arbitration latency which is needed to resolve the competition between multiple masters for the bus access. When one master is given the access, other masters must wait till the bus completes its current transaction. The wait time is the arbitration latency.

In this paper, priority arbitration mechanism and round robin mechanism are being implemented to measure the AHB latency. To measure the latency, in existing verification environment, monitors and scoreboards were added, to note down the request and grant signal along with time and number of clock cycles were calculated by subtracting the grant time and request time which gave us the exact arbitration latency for each transaction for both the arbiters.

The above graph is the analysis of results obtained after simulation which were run through a python script to plot the data in terms of a graph, plotted as latency versus simulation time. In Priority arbitration mechanism, the graph shows Device 8 has the highest priority hence low latency but devices 3, 2, and 1 are low priority devices which need

ISBN: 978-988-14047-9-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

to wait for the bus for a long time. The average wait time for priority arbitration was noted to be, 6 clock cycles.

Depending on the application and need the device priority can be changed so that devices which needs to be serviced will get the bus, a shared resource. The most common arbitrator used is round robin and priority can be incorporated in that as priority round robin arbitrator. The next analysis made is by replacing arbitrator to round robin to analyze AHB latency performance

Round robin arbitration is the most commonly used and fair method to provide equal access or share to all the masters to use the bus, either to send data to the slave or to read data from the slave. Round robin gives good predictability and performance when the masters have same bandwidth requirement. From the above graph, it is proved that average latency comes down to 2 clock cycles from 6 compared to priority but the bus is handled to all the masters whether the masters need it or not. The priority method could be unfair because the same master requester can get back to back bus access even the other master requesters are waiting, whereas round robin is a fair mechanism when the input stimuli has less bandwidth.

C. Bandwidth comparison between Priority and Round Robin arbitration schemes.

Bandwidth measures the rate at which data is sent across the bus typically in MB/sec. From the graph, it can be seen that the priority has high bandwidth of 8MB/sec and round robin has a lower bandwidth of 3MB/sec without utilizing the bus to its maximum efficiency. To have distributed bandwidth across the master one would prefer round robin, but in reality, bandwidth will not always be same and varies according to the application. So, to utilize the bus to its maximum efficiency one would prefer priority arbitration though the latency seems to be high which can mean the bus is kept busy by some master and other low priority masters are waiting to get on the bus.

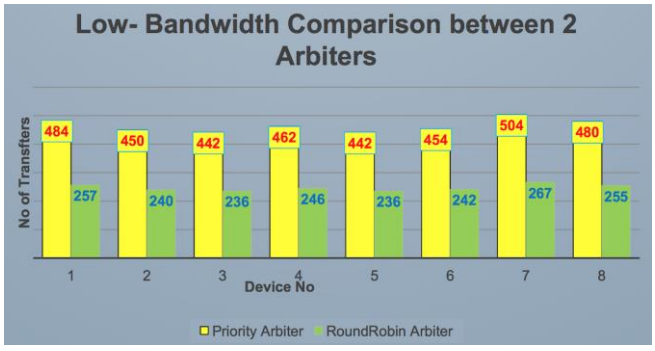

Fig. 2 Low Bandwidth comparison between 2 arbitrators

VI. Conclusion

The bus system preferred ideally involves a low latency and high bandwidth requirement. It can be seen that high latency for certain masters shows the bus is busy accessing the data from memory for the slave, indicating high bus utilization rate though latency for other masters may be high. The observation made was, high latency does not imply a low performance AHB bus or for that matter any communication bus. With latency going low, the most common fair arbitrator preferred for low and equal bandwidth data transmission is Round robin. For complex waveform generation and other communication applications the bandwidth requirement will be high. Depending on the application one may go for different design alternatives involving bandwidth arbitrator or priority or priority round robin, to end up having the application specific best alternative.

VII. Future Work

In SOC design it is always important to have good on chip bus architecture which integrates all the heterogeneous components on chip into a system. It also impacts system performance, power and area. For high bandwidth interconnect AMBA has AHB/ASB and for low peripherals, APB (Advanced Peripheral Bus). Once the evolution started in mobile and smartphones with multi cores, AXI (Advanced Extensible Interface) became popular which is a point to point interconnect which overcame shared bus constraints. To improve upon the current AHB design of this project, one can go design a reconfigurable arbiter and explore this space of possible configurations and custom tune it to high bandwidth, low latency and power effective arbitration system for AHB on chip bus.

Acknowledgement

We want to thanks Professor Morris Jones, San Jose State University, for providing us with UVM environment for AHB test bench.

References

[1] R. Sinha " The AMBA SOC platform", Springer, pp. 11-14
[2] Soren Sonntag, Helmut Reinig, "An efficient weighted-round-robin algorithm for multiprocessor architectures", IEEE, 22nd April 2008, Submitted for publication.
[3] Palakeeti Naveen Kalyan and K Jaya Swaroop, "Verification of AMBA-AHB based verifying IP using UVM methodology" International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD), Vol.5, Issue 4, Aug 2015, Submitted for publication
[4] G.S. Arunkumar, N. Soundar and D. Janarthanan, "Design of an efficient finite state machine for an implementation of AMBA AHB master" International Journal of Advanced Engineering and Recent Technology, Volume 3 Issue 1, January 2016 Submitted for publication.
[5] Kajol Singh, Shefali Verma and Shoba Sharma, "Design and implementation of multiple- master, multiple-slave AMBA AHB protocol Block with and without split/retry transfer for advanced microcontroller in Verilog/VHDL" IJECT Vol. 7, Issue 2, April - June 2016, submitted for publication.
[6] Soo Yun Hwang, Hyeong Jun Park, Kyoung Son Jhang, "An implementation and performance analysis of slave – side arbitration schemes for the ML-AHB bus matrix", Mobile Telecommunication Research Mobile, March 2007, submitted for publication.
[7] Amin M.A. El-Kustaban, Abdullah A.K. .Qahtan, "A Bus Arbitration Scheme with an efficient Utilization and Distribution", Internation Journal of Advanced Computer Science and Applications, Vol 8, No.3, 2017, submitted for publication.
[8] Texas Instruments "Introduction to the controller area network", May 2016, http://www.ti.com/lit/an/sloa101b/sloa101b.pdf
[9] Rishabh Singh Kurmi, Shruti Bhargava, Ajay Somkuwar, "Design of AHB protocol block for advanced microcontroller", International Journal of Computer Applications, N0. 8, Vol 32, Oct 2011, submitted for publication.
[10] Ashwin.P.Patel, M. Bhankhariya, Jignesh .S. Prajapati, "An overview of Transaction level modelling (TLM) in Universal Verification Methodology (UVM)", Journal of Information, Knowledge and Research in Electronics and Communication Engineering, Vol 02, issue 02, October 13, submitted for publication.