Implementation of AMBA AHB protocol for Wide Narrow BUS-SLAVE combination using VHDL

Varsha vishwarkama[1] Abhishek choubey[2] Arvind Sahu[3]
M.Tech, Rkdf-Ist,Bhopal Hod (Ec) Rkdf-Ist,Bhopal Asst.Prof.Tieit,Bhopal

Abstract-
The Advanced Microcontroller Bus Architecture (AMBA) is an open System-on-Chip bus protocol for high-performance buses on low-power devices. In this paper we implement a simple model of AMBA and use model checking and theorem proving to verify latency, arbitration, coherence and deadlock freedom properties of the implementation. Typical microprocessor and memory verifications assume direct connections between processors, peripherals and memory, and zero latency data transfers. They abstract away the data transfer infrastructure as it is not relevant to the verification. However, this infrastructure is in itself quite complex and worthy of formal verification. The Advanced Microcontroller Bus Architecture1 (AMBA) is an open System-on-Chip bus protocol for high-performance buses on low-power devices. In this report we implement a simple model of AMBA and verify latency, arbitration, coherence and deadlock freedom properties of the implementation. The verification is conducted using a model checker for the modal **μ**-calculus **Lμ**, that
has been embedded in the HOL theorem prover [3]. This allows results from the model checker to be represented as **HOL** theorems for full compositionality with more abstract theorems proved in HOL using a formal model theory of **Lμ** that we have also developed [4].

Keywords: AMBA, VHDL, ASB, APB, DMA, EDAROM, RAM, System-on chip.

1. Introduction- AMBA is an open specification that specifies a strategy on the management of the functional blocks that sort system on chip (SoC) architecture. It is a high-speed, high-bandwidth bus that supports multi master bus management to get the most out of system performance.

The AMBA specification defines three buses:

- Advanced High-performance Bus (AHB): The AHB is a system bus used for communication between high clock frequency system modules such as processors and on-chip and off-chip memories. The AHB consists of bus masters, slaves, an arbiter, a signal multiplexor and an address decoder. Typical bus masters are processors and DMA devices.
- Advanced System Bus (ASB): The ASB is also a system bus that can be used as an alternative to the AHB when the high-performance features of AHB are not required.

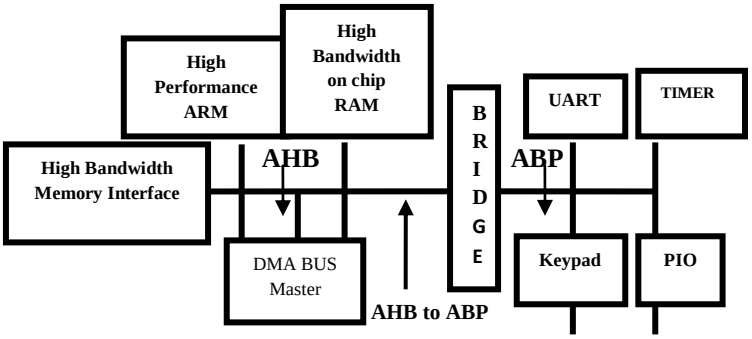

Fig 1 AMBA based microcontroller system

peripheral bus specialized for communication with low-bandwidth low-power devices. It has simpler interface and lower power requirements.

Designers can use either the AHB or the ASB in conjunction with the APB. The APB has a single bus master module that acts as a bridge between the AHB or ASB and the APB. The AMBA specification is hardware and operating system independent and requires very little infrastructure to implement. Figure 1 shows a typical AMBA-based microcontroller. We follow revision 2.0 of the AMBA specification.

2 AMBA APB- The APB is optimized for low power consumption and low interface complexity. It is used for connecting the high-band with system bus to low-bandwidth peripherals such as input devices. There is a single bus master, a single global clock and all transfers take two cycles. The bus master also acts as a bridge to the system bus, to which it can reconnect as a slave. The address and data buses can be up to 32 bits wide.

2.1 AMBA APB Specification-The operation of the APB consists of three stages, all of them are triggered on the rising edge of the clock:

1. IDLE. This is the initial and the default state of the bus when no transfer is under-way.

2. SETUP. The first stage of a transfer is a move to the SETUP state. The address, data and control signals are asserted during this phase but may not be stable. This stage always lasts for one clock cycle and then the operation moves to the ENABLE stage.

3. ENABLE. The address, data and control signals are stable during this phase. This phase also lasts one clock cycle and then moves to the SETUP or the IDLE stage depending on whether or not another transfer is required.

3 AMBA AHB- The AHB is a pipelined system backbone bus, designed for high-performance operation. It can support up to 16 bus masters and slaves that can delay or retry on transfers. It consists of masters, slaves, an arbiter and an address decoder. It supports burst and split transfers. The address bus can be up to 32 bits wide, and the data buses can be up to128 bits wide. As before, there is a single global clock. We choose to model the AHB rather than the ASB because the AHB is a newer design and also because it has been designed to integrate well with the verification and testing work flow.

3.1 AMBA AHB Specification-The operation of the AHB is too complex to be specified in terms of a few fixed stages. A simple transfer might proceed as follows (the list numbering below is not cycle accurate):

1. The AHB is in the default or initial state. No transfer is taking place, all slaves are ready and no master requires a transfer.
2. Several masters request the bus for a transfer.
3. The arbiter grants the bus according to some priority-scheduling algorithm.
4. The granted master puts the address and control information on the bus.
5. The decoder does a combinatorial decode of the address and the selected slave samples the address.
6. The master or the slave put the data on the bus and it is sampled. The transfer completes.

3.2 Granting bus access-The arbiter indicates which bus master currently the highest priority is requesting the bus by asserting the appropriate HGRANTx signal. When the current transfer completes, as indicated by HREADY HIGH, then the master will become granted and the arbiter will change the HMASTER **[3:0]** signals to indicate the bus master number. The arbiter changes the HGRANTx signals when the penultimate (one before last) address has been sampled. The new HGRANTx information will then be sampled at the same point as the last address of the burst is sampled.

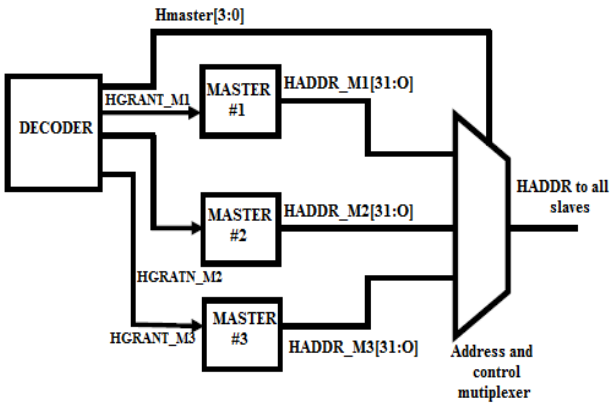

Fig. 3.1 HGRANTx and HMASTER signals are used in a system.

metal and the use of large on-chip memory blocks (such as Embedded DRAM) are driving factors which encourage the use of wider on-chip buses. Specifying a fixed width of bus will mean that in many cases the width of the bus is not optimal for the application. Therefore an approach has been adopted which allows flexibility of the width of bus, but still ensures that modules are highly table between designs.

The protocol allows for the AHB data bus to be ***8, 16, 32, 64, 128, 256, 512 or 1024***-bits wide. However, it is recommended that a minimum bus width of 32 bits is used and it is expected that a maximum of 256 bits will be adequate for almost all applications. For both read and write transfers the receiving module must select the data from the correct byte lane on the bus. Replication of data across all byte lanes is not required.

3.4 AMBA AHB signal list -This section contains an overview of the AMBA AHB signals. All signals are prefixed with the letter H, ensuring that the AHB signals is differentiated from other similarly named signals in a system design.

Name	Source	Description
HCLK Bus clock	Clock source	This clock times all bus transfers. All signal timings are related to the rising edge of HCLK.
HRESETn Reset	Reset controller	The bus reset signal is active LOW and is used to reset.
HADDR [31:0]	Master	The 32-bit system addresses bus.
HTRANS [1:0]	Master	Indicates the type of the current transfer, which can be..
HWRITE Transfer direction	Master	When HIGH this signal indicates a write transfer and when LOW a read transfer.
HSIZE [2:0] Transferse	Master	Indicates the size of the transfer, which is typically byte (8-bit), halfword (16-bit) or word (32-bit).
HBURST [2:0]	Master	Indicates if the transfer forms part of a burst.
HPROT [3:0]	Master	The protection control signals provide additional information
HWDATA [31:0] Write data	Master	The write data bus is used to transfer data from the bus master to the bus slaves during write operations.
HSELx	Decoder	Each AHB slave has its own slave

Slave select		select signal and this signal indicates that the current
HRDATA	Slave	The read data bus is used to transfer data from bus slaves to the bus master during read operations.
HREADY Transfer done	Slave	When HIGH the HREADY signal indicates that a transfer has finished on the bus.
HRESP [1:8]	Slave	The transfer response provides additional information on the status of a transfer.

3.5 Implementing a wide slave on a narrow bus-A wide slave being implemented on a narrow bus. Again only external logic is required and hence predesigned or imported blocks can be easily modified to work with a different width of data bus. Bus masters can easily be modified to work on a wider bus than originally intended, in the same way that the slave is modified to work on a wider bus, by:

• Multiplexing the input bus

• Replication of the output bus

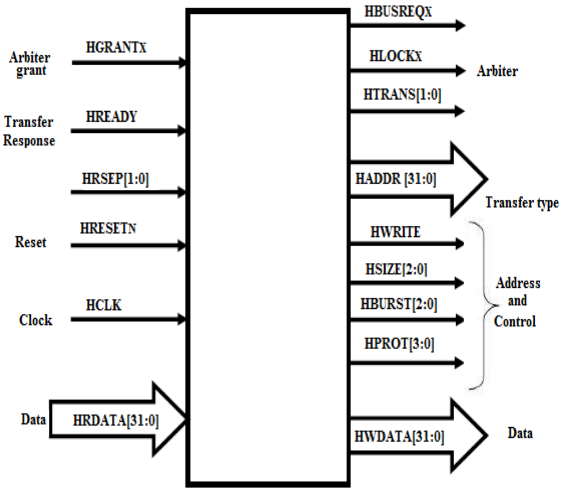

Fig.3.2 AHB bus master interface diagram

3.6 Implementing a narrow slave on a wider bus-A slave module, which has been originally designed to operate with a 32-bit data bus, can be easily converted to operate on a wider 64-bit bus. This only requires the addition of external logic, rather than any internal design changes, and therefore the technique is applicable to hard macrocells.

4 Results On Modelsim Using Xilinx Ise And Vhdl - The overall coding part can be writing on VHDL and simulate on ModelSim

4.1 Simulation Result of Slave 1-This simulation result contains the signal haddr =00100000, which shows that the master1 sending the address and control signals on the bus after the rising edge of the clock.Also the signal hlock1=1,the master1 requires locked access to the bus and no other master should be granted the bus until this signal is low.

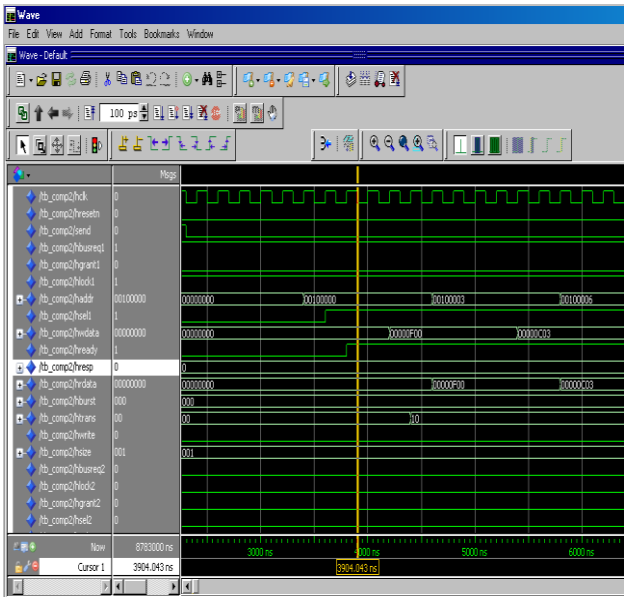

Fig.5.4 Simulation Result for the selection of slave1 and data transfer

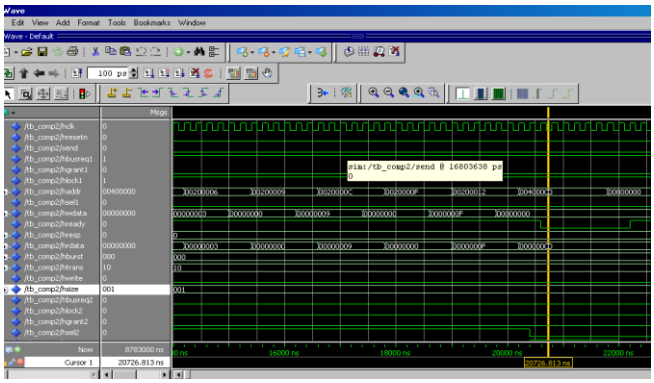

Fig.5.7 Simulation Result for wait state when hready=0

5. CONCLUSION - In this thesis we observe that the data transfer operation from one memory to another memory is fast as compared to serial communication by proposing the parallel communication in **AMBA AHB**. It also provides the opportunity to use master and slave up to 16 nos. and the data of every master is read and write simultaneously. In this implementation delay period is 4.33 ns and the clock period is 8.66 ns and frequency increases up to 115.401MHz..Here **AMBA AHB** supports the data transfer by reducing the time and increases the frequency of the bus to increase the system performance. The use of high capacity memory management with the **AMBA AHB** in this thesis successfully attempted to find the software solution for the problem of memory compliant in the microcontroller. The proposed implementation is capable of running in any PC with Xilinx and Modelsim EDA tools and FPGA board. This implementation able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other direct memory access devices reside. This implementation supports external memory up to 2GB.

REFERENCES –

[1] "AMBA specification (Rev2.0)", ARM Inc.

[2] Hu Yueli,Yang Ben "Building an AMBA AHB compliant Memory Controller" in 2011.

[3] Ashutoshku.Singh, Anurag Shrivastava, G.S.Tomar" Design and Implementation of High Performance AHB Reconfigurable Arbiter for on-chip Bus Architecture "in 2011.

[4] Wei Chipin,Li Zhaolin,Zheng Qingwei,Ye Jianfei, Li Shenglong"Design of Configurable Multichannel Interrupt Controller "in 2010.

[5] Yi Zhiqiang, Li Yun"On Chip Bus Design for HDTV SOC Decoder" in 2010.

[6] Ramesh Bhakthatchalu, Deepthy G R, Shanooja S. "Implementation of Reconfigurable Open Core Protocol Complaint Memory System using VHDL" in 2010.

[7] Zhichao Zhang, Wuchen Wu "UART integration in OR1200 based SOC design" in 2010.

[8] Purvi D. Mulani "SOC Level Verification Using System Verilog" in 2009.

[9] Guoling Ma, Hu He "Design and Implementation of an Advance DMA Controller on AMBA-Based SOC "in 2009.