



Design of AHB to Custom PIPE Bridge

Vyom Parikh¹, Nikunj Sanghani²

M.Tech [VLSI and Embedded Systems], Dept. of ECE, U.V.Patel College of Engineering, Ganpat University, Gujarat,
India¹

ASIC Engineer, Arastu Systems, Ahmedabad, Gujarat, India²

ABSTRACT:As the complexity of design of chip is increased in last decade, Chip execution has enhanced quickly these years. Conversely, memory latencies and data transfer capacities have enhanced little. As chip execution has enhanced, it has turned out to be progressively imperative to give a high-transfer speed, low-latencies memory subsystem to accomplish full execution capability of the processor. Memory controller deals with the execution of memory i.e. Memory controller manages the flow of incoming and outgoing data from the main memory. So there is always a requirement that Memory controller (MC) must understand the request given by the processor. AMBA AHB is extremely popular and stable and high-performance on-chip bus standard. The ultimate goal of this paper is to design a generic converter (Bridge) in terms of module which converts the request of AHB master BFM into suitable form which is easily understandable by the Memory Controller (MC). So that it can be easily used with memories like SRAM, DRAM, LPDDR etc.

KEYWORDS:AMBA, AHB, PIPE, Memory controller (MC), RTL (Register Transfer Level), BFM (Bus Functional Model)

I.INTRODUCTION

This paper describes the design of “AHB to Custom PIPE Bridge”. Advanced Microcontroller Bus Architecture (AMBA) – Advanced High-Performance Bus (AHB) is new generation AMBA bus.^[1] It describes an on chip communications standard for outlining high-performance embedded microcontroller. The AMBA AHB is for high-performance, high clock frequency system modules.^[1] The AHB acts as the superior framework backbone bus. AHB supports the systematic connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macrocell capacity. AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated test techniques.^[1]

The objective of this paper is to design a generic converter (RTL), which converts the request of the AHB master (BFM) into suitable form in terms of PIPE BFM with the help of FIFO so that the request can be easily understandable by the memory controller of the memories like DRAM, SRAM, LPDDR RAM etc.

Using AMBA-AHB bus, we can develop a generic converter for the memory controller which uses AHB slave. Our master can be a processor which gives request for data transmission to the module “Slave – PIPE RTL (AHB to custom PIPE Bridge)” which will convert the request in terms of PIPE interface which is easily understandable by MC and accordingly that request will be passed to the memory controller (MC). Memory controller takes care for the incoming and outgoing transmission to and from memory.

In this paper first block diagram of the design is shown and then specifications of design is described. Then Schematic view of design and synthesized view is shown. Simulation results are shown for selected operations.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

II.SPECIFICATIONS

A. Block Diagram of Module:

Fig. 1 is the block diagram for the AHB to custom PIPE Bridge. AHB Slave-PIPE Bridge is the main part of the project which is to be designed.

AHB Master (BFM) is the AHB master which sends request and control signals and data to be written to the memory.

AHB Slave-PIPE Bridge (RTL) is the main module of the project which is written into Verilog and responsible to translate the AHB slave transaction into custom PIPE interface and vice versa.

PIPE Interface (BFM) is the PIPE interface which is ready for the transmission when memory controller is ready.

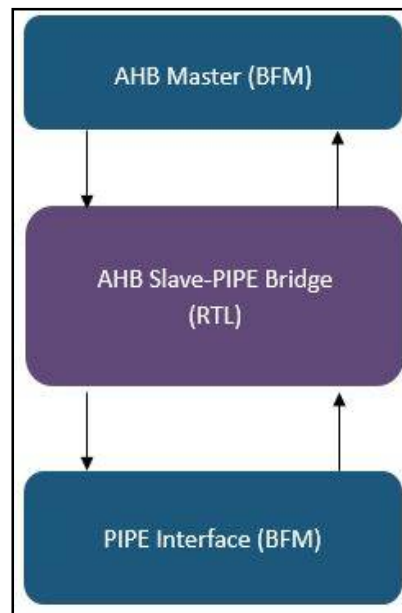


Fig.1 Block Diagram of Module "Slave-PIPE Bridge"

B. AHB Specifications

I. Microcontroller based on AMBA

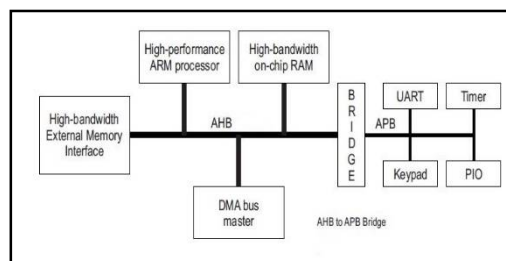


Fig. 2 Microcontroller based on AMBA

Microcontroller based on AMBA, usually encloses AHB, the peripheral memory bandwidth assisted by AHB, on which the microprocessor (CPU) and on-chip memory and supplementary DMA devices belong to.^[1] A high-bandwidth interface among the elements, taken part in the common transfers are supported by AHB bus.^[1] Also built on the high-performance bus, a connection to the lesser bandwidth APB, where maximum of the external devices in the system are located.^[1]

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

II. AHB

A new peers of AMBA bus is AHB. AHB assumes the supplies of high-performance synthesizable designs.^[1] It is a high - performance system bus that delivers high-bandwidth process and several bus masters are supported.^[1]

For high-performance and high clock frequency systems, AMBA AHB appliance the features including^[1]

- Transfer of Burst^[1]
- Split transaction^[1]
- Single-cycle bus master operation^[1]
- Single –clock edge Operation^[1]
- Non-tristate implementation^[1]
- Configuration of Data bus which is wider (64/128 bits).^[1]

A typical AMBA AHB system design contains the following components^[1]:

- **AHB master:** To start write and read operations adapted by a bus master, by adding control and address information. At a time, only one and only bus master is granted to strongly need the bus.^[1]
- **AHB slave:** Inside a specified address-space range, a bus slave acknowledge to a write or read operation. The success or waiting or failure of the data transfer, the slave send signals back to the active master.^[1]
- **AHB arbiter:** At a time, only one and only bus master is permissible to begin data transfers is assured by the bus arbiter.^[1] Despite the protocol of the arbitration is static, any arbitration algorithm, such as fair access or highest priority can be fulfilled conditional on the application requirements.^[1] Only single arbiter would include in AHB, although this would be insignificant in one bus master systems.^[1]
- **AHB decoder:** To decode the address and to give a select signal for a slave, of each transfer, which is involved in the transfer, the decoder is used.^[1] Only one centralized decoder is required in all AHB implementations.^[1]

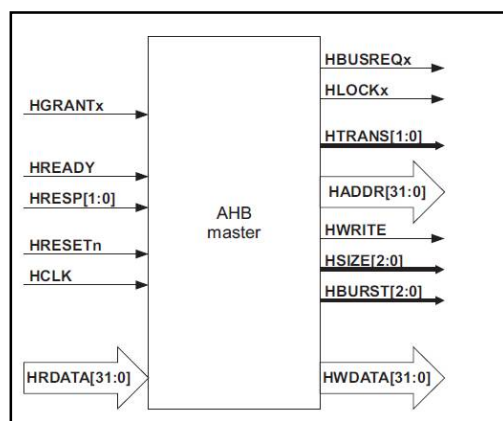


Fig. 3 Interface of AHB Master

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

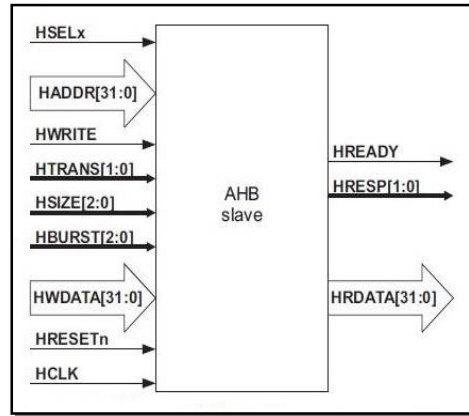


Fig. 4 Interface of AHB Slave

Fig. 3 shows the signals of AHB master and Fig. 4 shows the signals of AHB slave. Output signals from the master are input signals to the slave and vice versa. The basic overview of AHB operation are as discussed in the following section.

❖ Overview of AHB Operation

Before the AMBA AHB transfer starts the bus master must be allowed access to the bus.^[1] The process of allocation 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.^[1]

An allowed bus master starts an AMBA AHB transfer by driving the address and control signals.^[1] These signals give the information on the address, direction and width of the transfer.^[1]

A write data bus is used to move data from the master to a slave, while a read data bus is used to move data from a slave to the master.^[1]

Every transfer consists of:

- A control and address cycle
- For data, there is one or more cycle

The address cannot be extended and accordingly all slaves must sample the address during this time.^[1] The data, however, can be extended using the **HREADY** signal, when LOW this signal causes wait states to be inserted into the transfer and allows extra time for the slave to provide or sample data.^[1]

An AHB transfer consists of two distinct phases:

- The phase, which continues only single cycle called address phase.
- The phase, which may require more than one cycle called data phase. This is achieved using the **HREADY** signal

C. PIPE Interface BFM

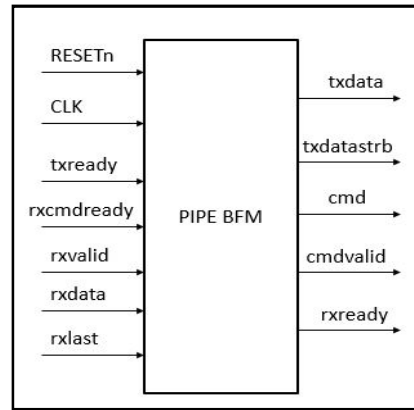


Fig. 5 PIPE Interface Signals

PIPE Interface is plug and play interface which can be used as interface to connect the different peripherals like USB, Memory controller of RAMs, etc. It is fully configurable.

❖ **Write and read operation functioning**

➤ **PIPE Write Operation**

- AHB Slave commence write request on PIPE interface if and only if txready Signal is asserted.
- At clock edge, cmdvalid signal is asserted along with write command and write address is given on txdata bus. txdatastrb is don't care for address transmission.
- From next clock edge AHB2PIPE interface sends data transaction on PIPE bus. AHB2PIPE should continue data transmission on PIPE interface once write transaction has initiated unless the end of the burst.

➤ **PIPE Read Operation**

- AHB Slave commence read request and cmdvalid is asserted and valid read command transmitted on PIPE interface along with read command. Read address is given on txdata bus with datastrb don't care.
- After accepting read request from slave AHB side, MC process read request and gives requested read data on rxdata&rxvalid. A valid data is received when rxready is asserted from AHB2PIPE, when rxvalid is asserted.

III. RESULT AND DISCUSSION

These simulations are done in Synopsis' VCS tool and its DVE wave viewer and design is compiled and synthesized in Synopsis' DC tool. In following Figures there are schematic view and simulations are shown. Fig. 6 describes the Schematic view of the module slave2PIPE (AHB to Custom PIPE Bridge). Fig. 7 describes the Write Operation, AHB master sends the write request by sending HWRITE = 1 and HSELx = 1 and other control signals are randomized. That request is translated in terms of MC and as shown in Fig. write operation would take place. Fig. 8 describes the Read Operation, AHB master sends the read request by sending HWRITE = 0, HSELx = 1 and other signals are randomized. On the other side, MC sends the data according to read request and length of transfer and according to that read data should pass on the HRDATA as shown in the Fig. 8.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

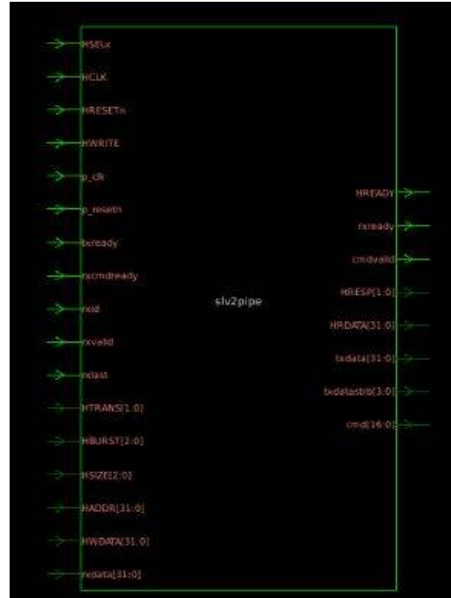


Fig. 6 Schematic View of AHB to Custom PIPE Bridge (Slve-PIPE RTL)

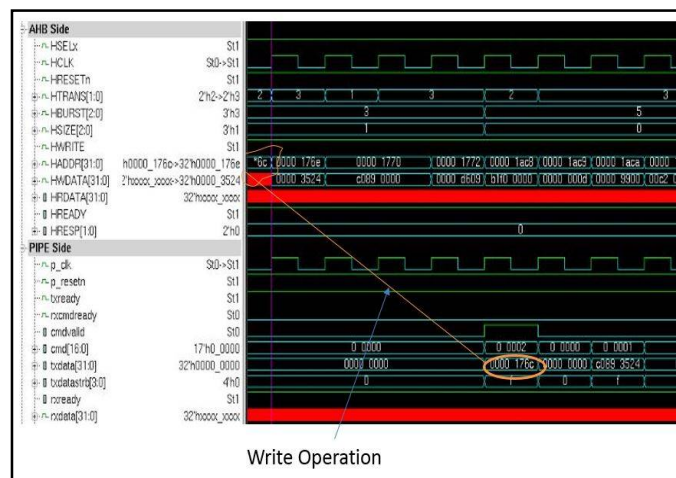


Fig. 7 Write Operation



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

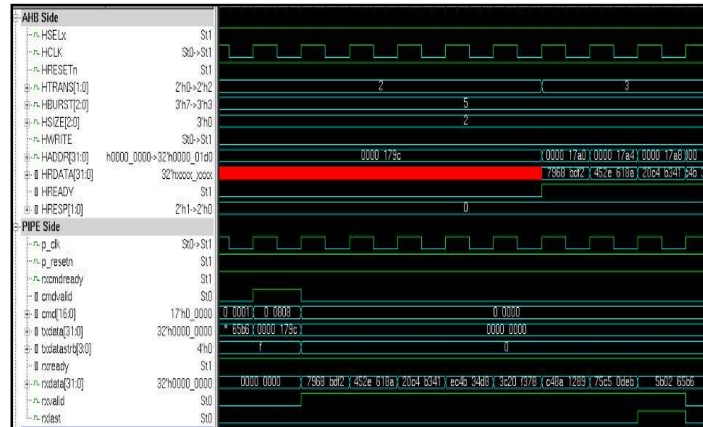


Fig. 8 Read Operation

IV. CONCLUSION

This Paper provides flexibility for the datawidth which is same on both side, i.e. input side (AHB side) as well as output side (PIPE interface i.e. MC side). Design Specifications are completely followed and verified. Simulation results shows that design works correctly.

REFERENCES

- [1] AMBA Specification 2.0 by ARM Inc. available at <http://soc.eecs.yuntech.edu.tw/Course/SoC/doc/amba.pdf>
- [2] "Overview of all specification available at <https://www.arm.com/products/system-ip/AMBA-specifications.php>
- [3] Wikipedia page available at <https://en.wikipedia.org/wiki/Advanced-Microcontroller-Bus-Architecture>
- [4] ARM Community Forum available at <https://www.arm.com/community/arm-connected-community-blogs-forums.php>
- [5] Samir Palintkar, Verilog HDL: A guide to Digital Design and Synthesis, 2nd Edition, Pearson Publication.