STUDY OF HIGH PERFORMANCE AMBA AHB RECONFIGURABLE ARBITER FOR ON-CHIP BUS ARCHITECTURE

1PRAVIN S. SHETE, 2SHRUTI OZA
1BVDU, 2Dept. of E & TC, College of Engineering, Pune, India
E-mail: shetepravin05@gmail.com, shruti.ozai1@gmail.com

Abstract- This paper focuses on study of Reconfigurable arbiter that can interface with any common IP core of a system, using specification of AMBA bus protocol. The arbiter plays a very important role to manage the resource sharing on the SOC platform. The scheme involves the typical AMBA features of ‘single clock edge transition’, ‘Split transaction’, ‘several bus masters’, ‘burst transfer’. The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. The reconfigurable arbitration algorithm, such as highest priority or fair access and round robin can be implemented depending on the application requirements.

Keywords- Reconfigurable Arbiter, Round Robin, Split Transfer, AMBA, IP)

I. INTRODUCTION

The Advanced Microcontroller Bus Architecture (AMBA) is an open System-on-Chip bus protocol for high-performance buses on low-power devices. 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 to 128 bits wide.

Arbitration to choose the next bus master uses a round robin algorithm. This ensures that no master gets starved. When a master has locked the bus, the round robin arbitration is overridden and the master with the lock retains highest priority to the bus. The sixteen AMBA BUS master0 through Master Slave0 through slave 15 are the sixteen AMBA Bus slaves. The AMBA AHB Bus Arbiter/Decoder contains a default master-master ( ), and a default slave – slave( ). AMBA AHB Bus Arbiter features are summarized.

1. AMBA AHB Bus arbiter function.
2. Round robin arbitration.
3. Default master-master ( ).
4. Default slave- slave ( ).

With the increasing number of system components in SOC design, it becomes that an efficient arbiter is one of the most critical factors for high system performance. Most existing buses have their own special protocols. The communication architectures defined by commercial standards are widespread and available in the market.

For example, the PI-Bus of OMI, the AMBA bus of ARM, the FISP bus of Mentor Graphics, the Core Connect of IBM, the Silicon Backplane of Sonics, the Wishbone of Silicione and others.

The rest of this paper is organized as follows:
Section II describes about Bus Specification of AMBA 2.0. Now in section III it describes typical AMBA bus Architecture. Section IV has AHB Arbiter Architecture which is typically used for Resource sharing.

Then section V shows the design description of AHB Arbiter and Last Conclusion of this paper.

II. BUS SPECIFICATION

AHB (AMBA High-performance Bus) is a bus protocol introduced in AMBA specification version 2 published by ARM limited Company. In addition to previous release, it has the following features:

- Single edge clock protocol.
- Split transaction.
- Several BUS Master.
- Burst transfers.
- Pipelined operations.
- Single cycle bus master handover.
- Non-tri-state implementation.
- Large bus-width (64/128 bit).

III. AMBA BUS ARCHITECTURE
Study of High Performance AMBA AHB Reconfigurable Arbiter for On-Chip Bus Architecture

Figure 1: Typical AMBA Bus Architecture [1].

The above figure 1 shows typical AMBA BUS Architecture which has three distinct buses under specification:

1. Advanced Peripheral bus (APB).
2. Advanced System Bus (ASB).
3. Advanced High Performance Bus (AHB).

1. Advanced Peripheral bus (APB).

The APB is part of the AMBA hierarchy of buses and is optimized for minimal power consumption and reduced interface complexity. The AMBA APB appears as a local secondary bus that is encapsulated as a single AHB or ASB slave device. APB provides a low-power extension to the system bus which builds on AHB or ASB signals directly. The APB Bridge appears as a slave module which handles the bus handshake and control signal retiming on behalf of the local peripheral bus.

The AMBA APB should be used to interface to any peripherals which are low bandwidth and do not require the high performance of a pipelined bus interface. All other modules on the APB are APB slaves.

The APB slaves have the following interface specification:

- Address and control valid throughout the access (un-pipelined).
- Zero-power interface during non-peripheral bus activity (peripheral bus is static when not in use).
- Timing can be provided by decode with strobe timing (un-clocked interface).
- Write data valid for the whole access (allowing glitch-free transparent latch implementations).

2. Advanced System Bus (ASB).

ASB is the first generation of AMBA system bus. ASB sits above the current APB and implements the features required for high-performance systems including:

- Burst transfers.
- Pipelined transfer operation.
- Multiple bus masters.

A typical AMBA ASB system may contain one or more bus masters. However, it would also be common for a Direct Memory Access (DMA) or Digital Signal Processor (DSP) to be included as bus masters. The external memory interface, APB Bridge and any internal memory are the most common ASB slaves. An AMBA ASB system design contains the following components:

- **ASB master**: A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.

- **ASB slave**: A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.

- **ASB decoder**: The bus decoder performs the decoding of the transfer addresses and selects slaves appropriately. The bus decoder also ensures that the bus remains operational when no bus transfers are required. A single centralized decoder is required in all ASB implementations.

- **ASB arbiter**: The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as highest priority or fair access can be implemented depending on the application requirements.

3. Advanced High Performance Bus (AHB).

AHB is a new generation of AMBA bus which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation. AMBA AHB has following features:
- Burst transfers.
- Split transactions.
- Single-cycle bus master handover.

- Single-clock edge operation.
- Non-tri-state implementation.
- Wider data bus configurations (64/128 bits).

Figure 2 shows the working principle of AHB Bus and also the bridging with lower level APB Bus. Bridging between this higher level AHB bus and the current ASB/APB can be done efficiently to ensure that any existing designs can be easily integrated. A typical AMBA AHB system design contains the following components:

- **AHB master**:
  A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.

- **AHB slave**:
  A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.

- **AHB arbiter**:
  The bus arbiter ensures that only one bus master at a time is allowed to initiate data transfers. Even though the arbitration protocol is fixed, any arbitration algorithm, such as highest priority or fair access can be implemented depending on the application requirements.

- **AHB decoder**:
  The AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.

A. When to use AMBA AHB/ASB or APB?

a) A full AHB or ASB interface is used for:
- Bus masters.
- On-chip memory blocks.
- External memory interfaces.
- High-bandwidth peripherals with FIFO interfaces.
- DMA slave peripherals.
- Very low power interfaces where clocks cannot be globally routed.
- Grouping narrow-bus peripherals to avoid loading the system bus.

IV. AHB ARBITER ARCHITECTURE

Figure 3 below shows AHB Arbiter Architecture which has several subsystem blocks. Few of them are controlled by the controller. Following are the different data path with different functionality:

![Figure 3: AHB Arbiter Architecture](image_url)

1. Priority logic block:
   The priority storage block is implemented through FSM approach. The priority scheme follows the round robin theorem of priority. The bus request have highest priority will get Grant First and Rest of request will wait for their priority.

2. Priority Storage block:
   The priority storage block is responsible to store the priority level. IT is the block which is responsible to enable the Bus req. Block and Interface Block. Grants are the input pin of this block.

3. Mux arrangement for grant and bus request:
   16:1 MUX for both grant and request.

4. OR gate:
   This block contains two sets of OR gate, each set contains 16 OR gate. First OR gate set is used to OR the output of each individual priority logic block ,second one is used to OR the individual grant form each priority logic.

5. Counter.

V. DESIGN DESCRIPTION

The following are the major steps involves in the Arbiter Design from architectural or functional point of view. The backbone architecture is shown in fig. 3 and is self-explanatory as per the clear depiction of the blocks.

1) The bus requests of different masters have to pass through the Bus req. block. This is responsible to pass the request to other logical blocks. This block depends upon the enable3 pin which is coming from Priority storage block.
2) Bus req. further passed through interface block and goes to priority logical block. The interface block is giving the enable signal; to priority logical block and interface block is also responsible for monitoring the data transaction through data_done signal; it can assert and de-assert the enable pin depends upon the data_done.

3) This bus_req. goes to the priority logic block; this block further decides that which master request will get the highest priority depending upon the priority. This block generates the grant signal.

4) This grant signal goes to priority storage block, encoder block and as output port to interact with Master. After getting the grant signal Master will send Address, Burst to indicate the type of transfer, and slave will also send Hready, Hresp and Hsplit.

5) At the same time when master samples the signal to the Arbiter. Grant signals which are the output of the Arbiter pass through the mux, inside the Arbiter, for this mux ‘Bus Master no’ is a select line, which indicates that which master is accessing the bus.

6) The output mux then passes to the controller block which will generate the necessary signals for counter, i.e. the controller will control the operation of counter.

7) Here some modification has been done in the previous architecture that the selection of arbitration is based on the requirement of the IP interface.

8) The grant output from the priority block is OR and then sent to the priority storage block which will store the priority and pass the enable signal; to the next priority depending upon the grant value and the whole operation is repeated depending upon the transaction.

CONCLUSION

In this paper, the Advanced Microcontroller Bus Architecture (AMBA) 2.0, which is an open System-on-Chip bus protocol for high-performance buses on low-power devices designed by ARM in 1999 is studied. Also, AHB Arbiter which monitors the AMBA Bus for request and choose the master with highest priority request as the next AMBA bus transaction master. In the next phase, Design and Implementation of AHB Reconfigurable Arbiter using Verilog Hardware Description Language will be done.

REFERENCES

[5]. Guoliang Ma* and Ru Re “Design and Implementation of an Advanced DMA Controller on AMBA-Based SoC”, 2009 IEEE.
[7]. Chenghai Ma, Jinan,China, Zhijun Liu, Jinan,China, Xiaoyue Ma , “Design and Implementation of APB Bridge based on AMBA 4.0”, 2011 IEEE.
[8]. Technical data-sheet related to AHB to APB Bridge by CADENCE.

---

Study of High Performance AMBA AHB Reconfigurable Arbiter for On-Chip Bus Architecture

15