COMIT SYSTEMS
The Contract Engineering Company
Design Advantage®
Comit Systems, Inc. Newsletter
We build them for YOU.SM
Volume 5 Number 1 Jan-Feb 2002

Dealing with on chip memories and AHB

*By **Prashant Joshi**,* Comit Systems, Inc.

Introduction

The ARM® AMBA AHB is a widely preferred bus for System On Chip (SoC) applications, and has proven its performance with many complex chip architectures.

Developments in manufacturing techniques and innovations in deep sub-micron technology have made possible applications with large on-chip memories. Designs are targeted at increasingly higher frequencies as well. The number of masters and slaves on the AHB bus are also increasing and are becoming increasingly more complex.

While designing with the AHB bus, design, verification and synthesis must take care of certain issues, which, if not handled carefully, lead to higher NRE costs and re-spins. This article discusses a common challenge and some solutions.

About the AHB Bus

The AMBA AHB is a new generation AMBA bus, intended to address the requirements of high performance synthesizable designs. The AMBA AHB implements features required for high-performance, high clock frequency systems including:

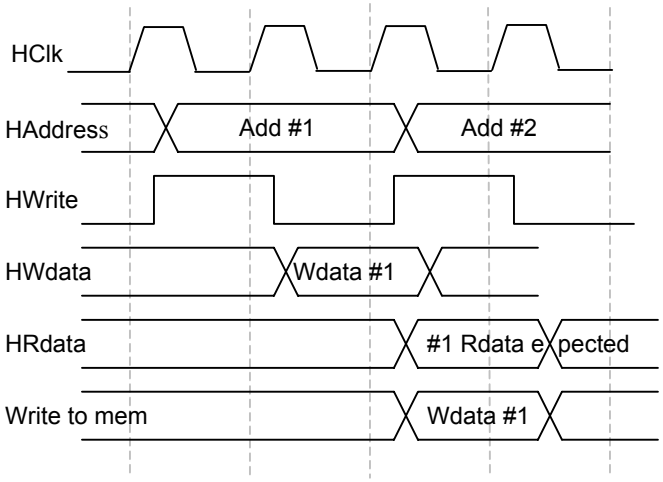
- Burst transfers
- Split transactions
- Single cycle bus master handover
- Single clock edge operation
- Non-tristated implementation
- Pipelined bus
- Wider data bus configurations (64/128 bits).

Typical AHB system components

The AHB bus can have multiple masters like DMA controller, Internal Memory controllers, system specific components like video processing engines, Ethernet controllers, wireless media access controllers, External Memory controllers, AHB to APB bus bridge for slower peripherals like timers and Interrupt controller. These components can be either master or slave on the AHB bus.

The Design Challenge

Let's take an example of an on chip common memory (SSRAM type) as a slave on the AHB bus. Different Masters on the bus will execute transfers which can be either single read, write, Read followed by a Read, Read followed by a Write, Write followed by a read, Write followed by a write.



Address and control are available in the first clock cycle. Since write data is available at the second clock cycle, actual write to the memory location addressed by the Add #1 will happen only during the third clock cycle. The second cycle is a read at the same address location; hence the address will not change. Because of the pipelined nature of the bus, the AHB master will expect read data after the third clock cycle. As the actual write to memory location 1 is happening in the third clock cycle, the slave will return the old data available at Add #1. Also, depending on the physical location of the memory block, RC delays could be large, creating timing problems in meeting the single cycle access.

Solutions

One can address this problem in several different ways while implementing the AHB slave. A few are listed below.

Case 1: Default wait state insertion

By keeping track of the transfer cycles, a default wait state can be added in the slave to respond to a read after a write access. In other words, a read that follows a write, will have one wait state access. This solution is good for FPGA implementations where an ARM test chip is on board and all slaves are implemented in the FPGA: in design prototyping prior to building an ASIC, for example.

Disadvantage: This solution has a disadvantage: every read, that follows a write cycle, will undergo a wait state, although access may not be to the same address. If the write and successive read are at two different addresses, this will degrade bus performance.

Case 2: Multiplexer implementation
A multiplexer can be implemented in the read path to multiplex read data from the slave and the write data bus. When the access is a read access at the same address location after a write, one needs to enable the multiplexer, which will transfer a write data back on the read data bus.

Disadvantage: This method, although it returns the data written in the first cycle back in the successive read cycle (thus saving a clock cycle in this kind of access) does not guarantee that the data is actually written in the memory. It includes additional multiplexer delays in the data path as well. To verify that the data is actually written in memory, a separate access to the same memory location will be required.

Case 3: Selective Wait insertion
One can keep track of successive transfer cycles by registering the previous access address and the type of access. If the previous transfer address and current transfer address matches, and if previous cycle was a write cycle and the current cycle is a read cycle then one needs to insert a wait state.

Disadvantage: This method requires actual address comparison, which will increase the gate count of logic per slave implementation.

By understanding the performance requirements of each master & slave and calculating the AHB bus throughput, appropriate solutions may be adopted by the designer, as needed. ■