International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-3, June 2015
ISSN: 2395-3470
www.ijseas.com

Design & Implementation of Advance Peripheral Bus Protocol

Heli Shah[1], Chinmay Modi[2], Bhargav Tarpara[3]
[1, 2,] *M.Tech. VLSI, U.V.Patel College of Engineering and Technology, Kherva, Mehsana, India*
[3]*Verification Technical Assistant, eiTRA, Ahmedabad, India*

***Abstract*- This design presents an intellectual property (IP) for inter-Advanced peripheral bus (APB) protocol. The current VLSI design scenario is characterized by high performance, complex functionality and short time-to market. A reuse based methodology for SoC design has become essential in order to meet these challenges. The work involved is of APB Protocol and its slave Verification. The idea behind this is to test DUT. Propose model is used for communication between master and slave. The entire design has been coded in Verilog & verified using Spartan kit.**

Index Terms— Intellectual property (IP), Advanced microcontroller bus architecture (AMBA), Advanced peripheral bus (APB), Field-programmable gate array *(FPGA), Verilog. Protocol, Data, Advanced high performance bus (AHB), Advanced system bus (ASB).*

I. INTRODUCTION

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) protocol family. It defines a low-cost interface that is optimized for minimal power consumption and reduced interface complexity. The APB protocol is not pipelined, use it to connect to low-bandwidth peripherals that do not require the high performance of the AXI protocol. It relates a signal transition to the rising edge of the clock, to simplify the integration of APB peripherals into any design flow. Every transfer takes at least two cycles.

The APB can interface with: (1) AMBA Advanced High-performance Bus (AHB), (2) AMBA Advanced High-performance Bus Lite (AHB-Lite), (3) AMBA Advanced Extensible Interface (AXI), (4) AMBA Advanced Extensible Interface Lite (AXI4-Lite).

APB Revisions are (1) AMBA 2 APB Specification (2) AMBA 3 APB Protocol Specification v1.0, (3) AMBA APB Protocol Specification v2.0. [1]

The AMBA 3 APB Protocol Specification v1.0 defines the following additional functionality:

• Wait states.
• Error reporting.

These are **PREADY** is a ready signal to indicate completion of an APB transfer. **PSLVERR** an error signal to indicate the failure of a transfer.

II. AMBA BUS

The AMBA specification are:

- This is development of embedded microcontroller products with one or more CPUs or signal processors.
- This is highly reusable peripheral appropriate for full-custom, standard cell and gate array technologies.
- It provides a road-map for advanced cached CPU cores and the development of peripheral libraries to minimize the silicon infrastructure required to support efficient on-chip.

An AMBA is having backbone bus AMBA AHB or AMBA ASB. It sustains external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access (DMA) devices abide. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also it is a bridge to the lower bandwidth APB. [1]

AMBA APB provides the basic peripheral macro cell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus. It consist of interfaces which are memory-mapped registers.

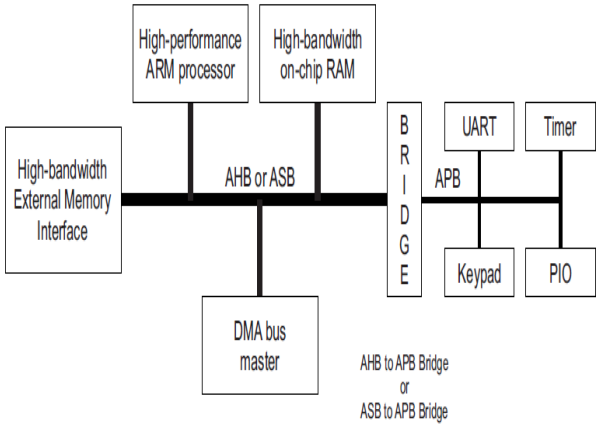

Figure 1 AMBA Bus Architecture [1]

No high-bandwidth interfaces. It has been accessed under programmed control. The external memory interface is application-specific and may only have a narrow data path, but may also support a test access mode which allows the internal AMBA AHB, ASB and APB modules to be tested in isolation with system independent test sets.[1]

A. AMBA AHB

Advanced High performance bus (AHB) is made for address the requirements of high-performance synthesizable designs. AMBA AHB is a new level of bus for the APB and implements the features high clock frequency systems including:

- burst transfers
- split transactions
- Single cycle bus master handover
- Single clock edge operation
- Wider data bus configurations (64/128 bits).

B.AMBA ASB

The Advanced System Bus (ASB) specification defines a high-performance bus that can be used in the design of high performance 16 and 32-bit embedded microcontrollers. It supports the efficient connection of processors, on-chip memories and off chip external memory interfaces with low-power peripheral macro cell functions. The bus also provides the test infrastructure for modular macro cell test and diagnostic access. [1]

C. AHB VS APB

AHB stands for Advanced High-performance Bus and APB sands for Advanced Peripheral Bus. Both the Advanced High-performance Bus and the Advanced Peripheral Bus are part of the Advanced Microprocessor Bus Architecture (AMBA). Though both the AHB and the APB belong to AMBA, they differ in many way.

Difference between the two, the AHB uses a full duplex parallel communication whereas the APB uses massive memory-I/O accesses. Both the AHB and the APB are on chip Bus standards. The Advanced High-performance Bus is capable of waits, errors and bursts. The ADH, which is pipelined, mainly connects to memories.

When comparing the usage, the APB is simpler than the AHB. Unlike the AHB, there is no pipelining in APB. The APB is mainly proposed for connecting to simple peripherals. Looking at the AHB and the APB, it can be seen that the APB comes with a low power peripheral.

It can also be seen that Advanced Peripheral Bus is sometimes optimized for reduced interface complexity and minimal power consumption for supporting peripheral functions. This Bus can also be used in union with either version of the system bus.

When looking at the features of AHB, it has a single edge clock protocol, several bus masters, split transactions, single-cycle bus master handover, burst transfers, large bus widths.

International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-3, June 2015
ISSN: 2395-3470
www.ijseas.com

In AHB, the transaction consists of an address phase and a data phase. In case of AHB, there is only one Bus master at a time.
When compared to Advanced High-performance Bus, the Advanced Peripheral Bus is only used for low bandwidth control accesses. Though the APB has an address phase and data phase as like that of the AHB, it comes with a list of low complexity signal. [4]

III APB BLOCK DIAGRAM

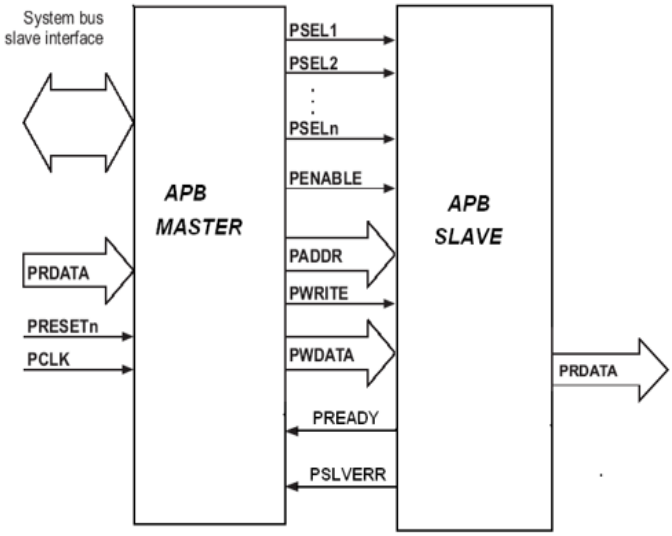

Figure 2 Interfacing of APB Master & Slave [2]

Table 1. List of APB signals [2][3]

Signal	Signal Description
PCLK	Clock. The rising edge of PCLK times all transfers on the APB.
PRESET	System bus equivalent Reset. The APB reset signal is active LOW.
PADDR	32 bit. address bus
PSEL	The slave device is selected and that a data transfer is required.
PENABLE	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
PWRITE	Access when HIGH.
PWDATA	32 bits. Write data .PWRITE is HIGH.
PREADY	Ready. To extend an APB transfer.
PRDATA	32 bits. Read data. PWRITE is LOW.
PSLAVERR	Slave error. This signal indicates a transfer failure,

A. Operating states of APB

IDLE is the normal state of the APB. When a transfer is necessary the bus relocates into the SETUP state, where the suitable select signal, **PSELx**, is asserted.

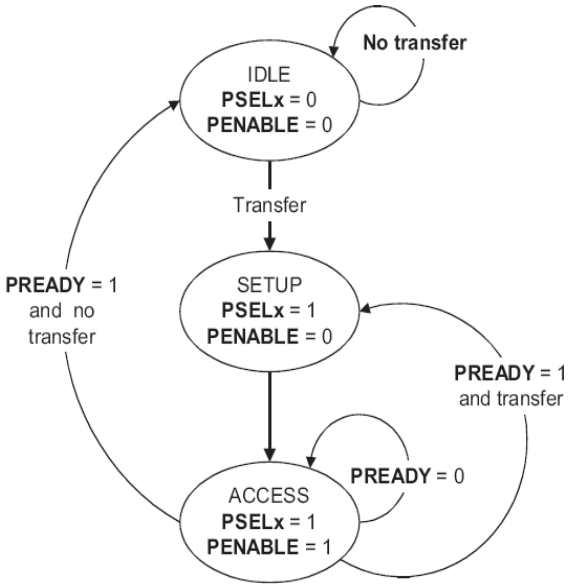

Figure 3 State diagram [2][3]

The bus only waits in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock. **ACCESS** will enable signal, **PENABLE**, is asserted in the ACCESS state. The write, write data signals, select, and address must remain stable during the transition from the SETUP to ACCESS state. ACCESS state is controls when to exit by the **PREADY** signal from the slave. These are the conditions one is if **PREADY** is

held LOW by the slave then the peripheral bus remains in the ACCESS state another is **PREADY** is driven HIGH by the slave then the ACCESS state is exited and the bus returns to the IDLE state if no more transfers are required after that it will start the same cycle. [2][3]

B. Write cycle

At T1, a write transfer starts with PADDR, PWDATA, PWRITE, and PSEL, being registered at the rising edge of PCLK. It is called the SETUP cycle. At the next rising edge of the clock T2 it is called ACCESS cycle, PENABLE, and PREADY, are registered. When asserted, PENABLE indicates starting of Access phase of the transfer. When asserted, PREADY indicates that the slave can complete the transfer at the next rising edge of PCLK. The PADDR, PWDATA, and control signals all remain valid until the transfer completes at T3, the end of the Access phase .The PENABLE, is disabled at the end of the transfer. The select signal PSEL is also disabled unless the transfer is to be followed immediately by another transfer to the same peripheral.[2][3]

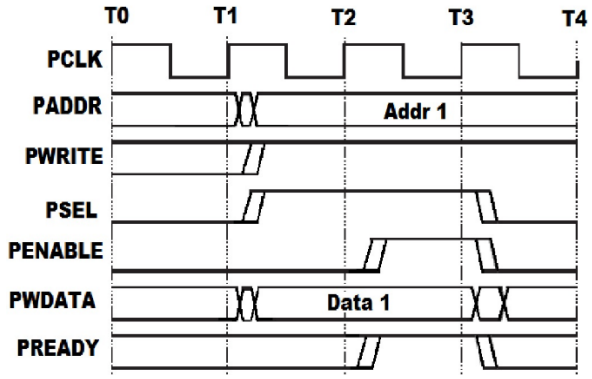

Figure 4 Write cycle

C. Read cycle

During read operation the PENABLE, PSEL, PADDR PWRITE, signals are asserted at the clock edge T1 (SETUP cycle). At the clock edge T2, (ACCESS cycle), the PENABLE, PREADY are asserted and PRDATA is also read during this phase. The slave must provide the data before the end of the read transfer.[2][3]

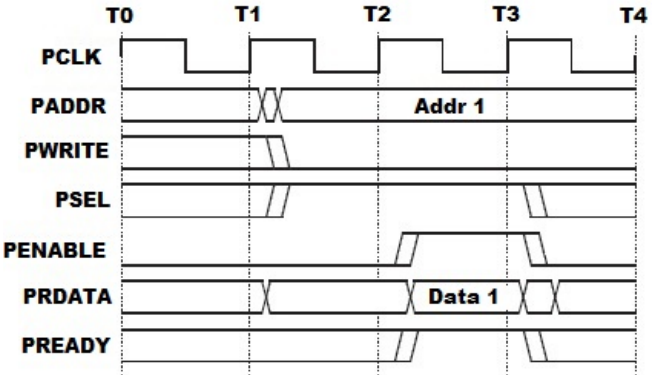

Figure 5 Read cycle

IV.SIMULATION RESULTS FOR DESIGN

Whenever clock signal goes high from an operator at that instance PENABLE and PREADY goes high PADDR which is of 32-bit in length. After enabling PADDR it will take data and write it on PWDATA and which is also transferred to the apb_write_data which all are 32 –bit in size

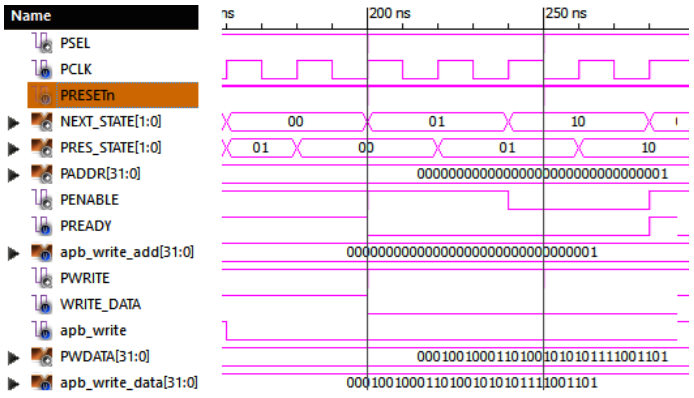

Figure 6 Simulation result of write cycle

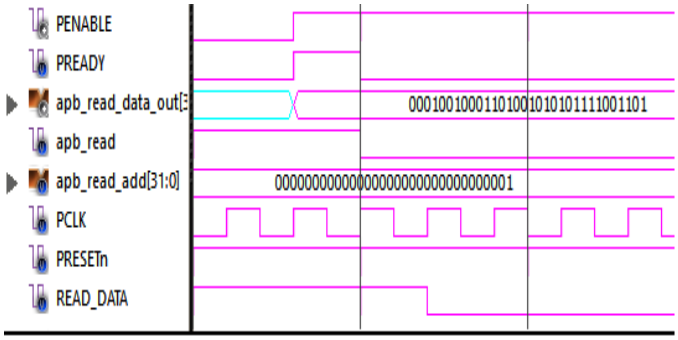

Figure 7 Simulation result of Read cycle

After transfereing the data on the adress and after ready toggle once with penable after this operation apb_read goes high and its reads the data from the apb_write _data and put it on apb_read_data.

V. CONCLUSION

This paper gives an outline of the AMBA bus architecture and explain the APB bus in detail. The APB bus is designed using the Verilog HDL according to the specification and is verified using Xilinx. The simulation results show that the data read from a particular memory location is same as the data written to the given memory location. Hence, the design is functionally correct. Xilinx also ensures the functional correctness of the design.

The electronic system level model of the same design will be created in the future since ESL is the requirement of the future because of increasing design complexity. The results obtained after the simulation will be compared with the results.

VI. MOTIVATIONAL WORK

Here we have designed Advanced Peripheral Bus as a single slave but as we know APB is a multiple slave interface so we can design that accordingly.

So After completion of this, we can move to physical design for IC fabrication for the same.

VII. REFRENCES

[1]URL:http://wwwmicro.deis.unibo.it/~magagni/amba99.pdf
[2]ARM, “AMBA Specification Overview”, available at http://www.arm.com/.
[3] ARM, “AMBA Specification (Rev 2.0)”, available at http://www.arm.com.
[4]URL:http://www.differencebetween.net/technology/difference-between-ahb-and-apb
[5] Samir Palnitkar, “*Verilog HDL: A guide to Digital Design and Synthesis (2nd Edition)*, Pearson, 2008.
[6] URL:http://www.testbench.com.