DESIGNING AND IMPLEMENTATION OF AMBA APB PROTOCOL

1 Mr.E.SAKTHIVEL,2 RAVI D,3 SIVASANKAR K ,
1Assistant Professor, 23UG scholars, Department of Electronics and Communication Engineering
Adhiyamaan College of Engineering (Autonomous), Hosur

Abstract

The APB (Advanced peripheral bus) protocol is a part of AMBA (advanced microcontroller bus architecture) family. Design under test (DUT) is tested and it establishes the communication between master (test bench) and slave (design). It is designed based on a reusable based methodology for system on chip (SOC) which is essential in order to meet the current VLSI challenges. APB involves low bandwidth, low cost and minimal power consumption and is used to connect the Timer, Keypad and other devices to the bus architecture. The work involved is of APB protocol design and implementation. Here we are designing six test cases namely single and multiple write transaction with and without wait, multiple write transaction with and without wait states, multiple read and write transaction with and without wait. The design is programmed using Verilog HDL and tested using verilog test bench. And finally the APB design is verified using QuestaSim tool.

Keywords : AMBA, APB Protocol, SOC, VLSI, Verilog HDL, Low Power Consumptions.

1.introduction

APB comes under AMBA 3 protocol family. AMBA specifications standards are used for designing high-level embedded microcontrollers and it provides independence in technology and to encourage the modular system design. An AMBA protocol strongly encourages to reuse peripheral devices to minimize the silicon infrastructure. There are lots of protocols that comes under AMBA protocol ex:(CHI,ACE,AXI,AHB,ASB,APB). The APB- Advanced peripheral bus protocol is actually a part of the Advanced Bus Architecture protocol family (AMBA). It is defined as a less priced interface because of its optimization for less consumption of power and less complexity of interface. APB (advanced peripheral bus) is non-pipelined, used for connection with low bandwidth peripheral which does not require the AXI (high performance) protocol. To simply the integration of APB peripherals into a design flow it relates a signal transition to the rising edge of the clock. Each transfer takes at least two Cycles (setup cycle and access cycle). It can be used for accessing the programmable control registers of the peripheral devices.The APB can be interfaced with AMBA AHB (Advanced high performance bus), AMBA AHBLite (Advanced high performance bus lite), AMBA AXI (Advanced extensible interface), AMBA AXI4lite (Advanced extensible interface lite)

1.1. APB block diagram

The advance peripheral bus protocol (APB) is designed as to the design specification.[1]. The following block diagram shows the basic signals of advance peripheral bus protocol in figure1.

The slave part of APB will takes the following signals PCLK, PWRITE, Present, PENABLE, PSEL as inputs and 32 bit PRDATA as output as shown in fig 1. PWDATA and PADDR is also 32 bit (input). The APB signals are explained in below table 1.

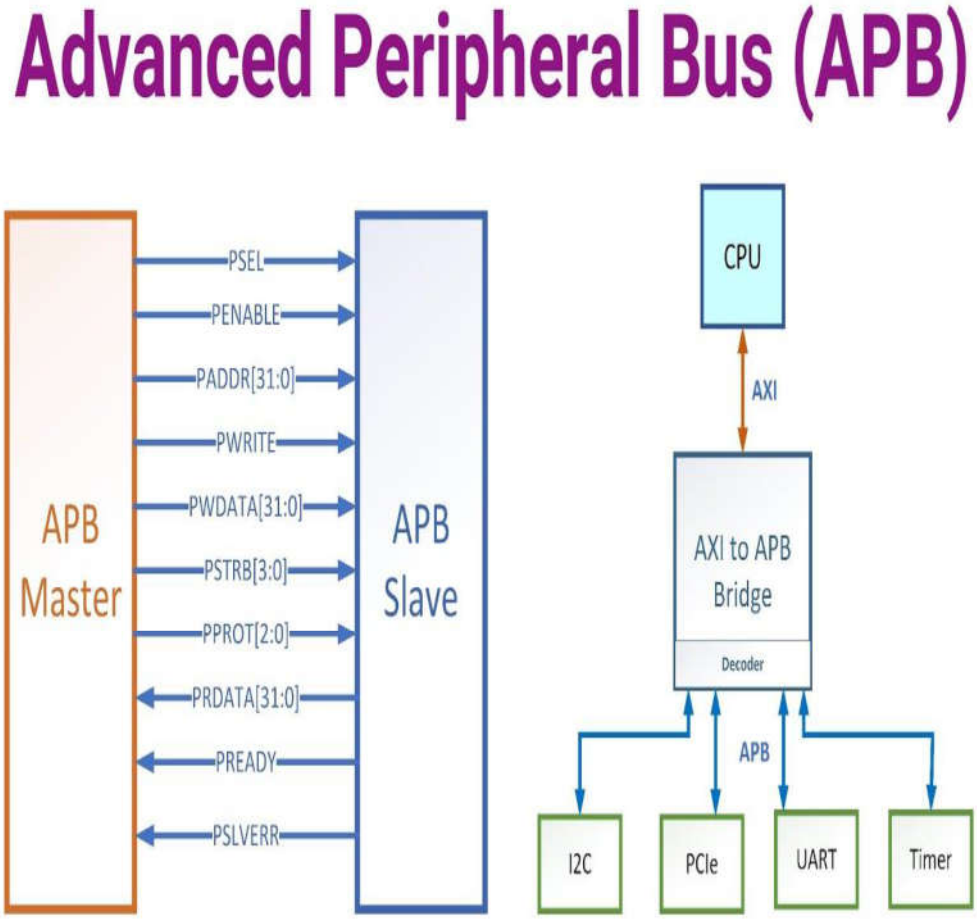

Figure 1. APB block diagram

Table 1. APB signals and functions of the signals.

S.NO	SIGNALS	SIGNALS NAME	SIGNALS FUNCTION

1.	K SETn	Clock signal, Reset signal	APB works along rising edge of PCLK. PRESETn signal to reset APB.
2.	PADDR	32 bit address	Location mapping
3.	PSELX	Select signal(control	Transaction starts
4.	PWRITE	Direction signal	PWRITE=1(writetransfer) PWRITE=0(read transfer)
5.	PENABLE,PREAD	Enable and ready Signal	To indicate transaction is completec
6.	PWDATA	Write data(32 bits)	Data is written in Particular adderss
7.	PRDATA	Read data(32 bits)	Read data will appears when we c particular address .

APB PROPOSED WORKING MODEL:

The operation of APB is done by using FINITE STATE MACHINE (fsm) [1]. There are totally three states namely, IDLE, SETUP and ACCESS states. In APB every transfers (read or write) takes at least two cycles (SETUP phase and ACCESS phase). The first transfer (either read or write) will take three clock cycles, but the following transactions are completed in two clock cycles [3]. The state transfer is shown in below figure 2.

2.1. Idle state

Idle state is also called as default state. This state indicates no operation is performed. All signals like PSEL, PENANBLE, PADDR will initially unknown (zero). If no transfer is required then automatically idle state will asserted.

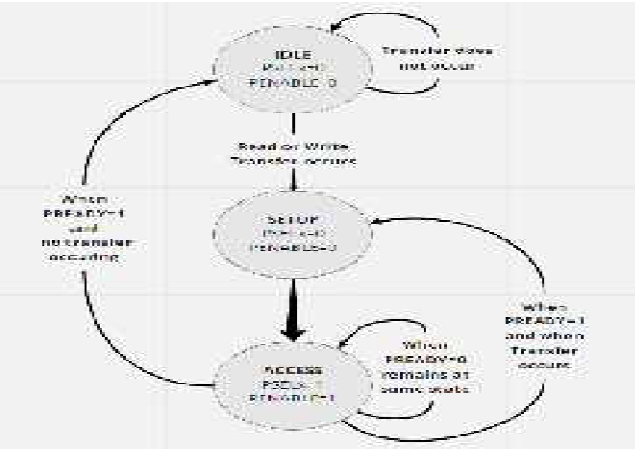

Figure 2. State diagram of APB

2.2. Setup state

Setup state is active when the transfer is required. In this setup phase, PSEL signal will asserted. This indicates beginning of transfer and the presence of PADDR and PDATA. The bus will enters setup phase only when the transfer is needed, otherwise bus will remains in idle phase. Also PWRITE, PADDR, PWDATA are provided in this phase. The bus will remain in setup phase for one clock cycle and on the upcoming positive clock or rising edge of clock, the bus will automatically moves to ACCESS state.

2.3. Access state

This state is mainly used to tell that the transaction is completed. At the starting of access state PENABLE will asserted high(1) and the remaining signals like(PADDR,PWRITE,PSEL,PWDATA)all are remains stable as in setup phase, no changes in this signals. In access state PENABLE and PREADY signal will asserted high (1) and this assertion of both PENABLE and PREADY indicates completion of one transaction. If further transfer needed, then bus will moves to SETUP phase otherwise it moves to the default state (IDLE STATE).

3.APB WRITE AND READ TRANSFER

3.1. Write transfer

In default the bus will remains as IDLE state in T0 clock. While write transaction needed then the signals PSEL, PWRITE, PADDR and PWDATA will asserted high(1) in T1 clock edge (setup phase). This setup phase indicate that the transaction is started.[4]. And the next rising edge of the clock T3, PENABLE and PREADY both are asserted high (1) figure 3, which indicates the completion of one write transaction in APB. As we mentioned earlier APB will take minimum two clock cycle to complete per transaction. If any further transaction needed then PENABLE signal will became zero and again it enters into SETUP state. At the same time PREADY signals moves from high to low for next write transaction

Write transfer with wait statement is to control the transaction. Here PENABLE signal will remains high until PREADY also asserted high. And if both this PENABLE and PREADY are asserted then only it moves to next write transaction. So for the first write transaction it will take minimum three clock cycle because of this wait condition figure 4. Once this condition finished then normally next write transfer will take only two clock cycles.

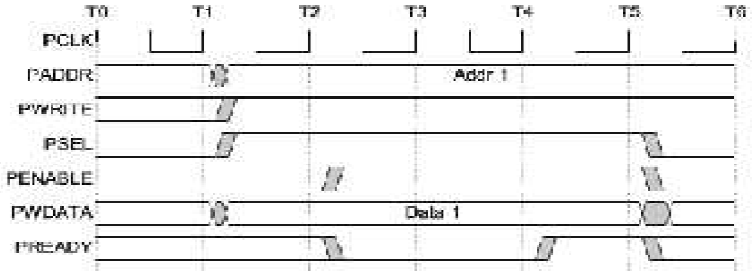

Figure 4. Write transfer with wait

.

Similar to write transaction, read transaction will work. PRDATA signal is output in nature, whenever the PADDR is presents the prdata presents in that particular address will updated .In default the bus will remains as IDLE state in T0 clock. While read transaction needed then the signals PSEL, PWRITE, PADDR will asserted high(1) and PWRITE signal asserted low(0) in T1 clock edge (setup phase). This setup phase indicate that the read transaction is started. And in the upcoming positive edge of the clock T3, PENABLE and PREADY both are became high (1), which indicates the completion of one read transaction in APB[2]. As we mentioned earlier APB will take minimum two clock cycle to complete per read transaction. If any further transaction needed then PENABLE signal will became zero and again it enters into SETUP state. At the same time PREADY signals moves from high to low for next read transaction figure 5.

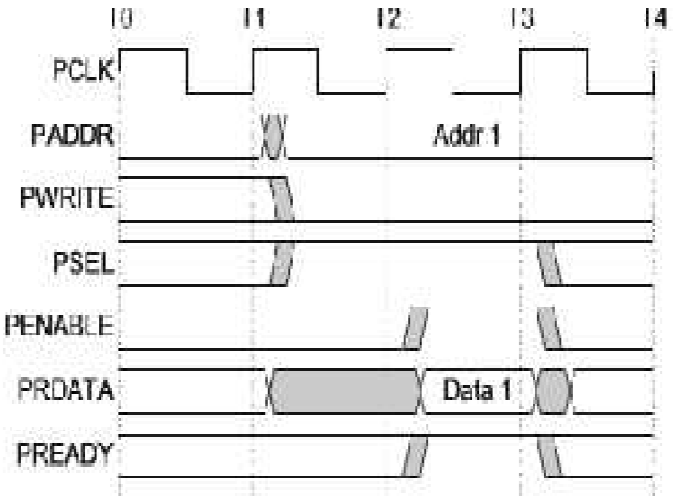

Figure 5. Read transfer without wait

3.2 Read transfer with wait.

To read the data that available in particular address PWRITE must be asserted zero. And to complete per read transfer PENABLE and PREADY both must be high in next clock cycle T3 figure 6. Here because of wait statement PENABLE will wait until PREADY also asserted high figure 6.

4. RESULTS AND DISCUSSION

The design and test bench part is programmed by using verilog HDL [5]. And compiled using QUESTASIM tool. Analysis is done for six different test cases.

4.1 Output for single write transaction without wait

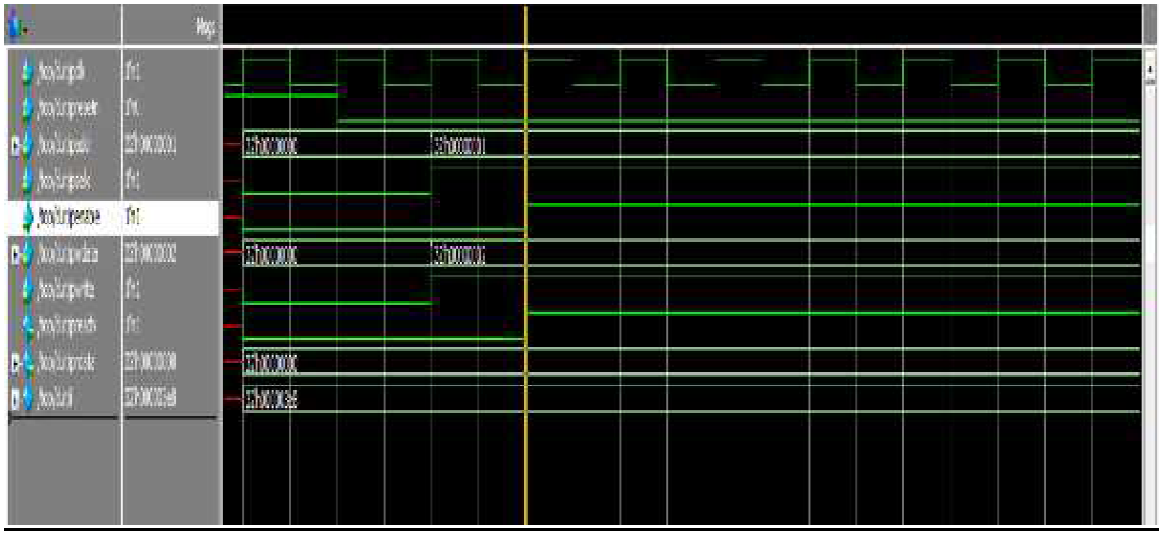

Figure 7.single write without wait

In above figure 6, single write transfer is obtained. Initially bus will present in idle state. So all singles are zero and in the next clock bus will move to setup state.

4.2. Output for multiple write transfer:

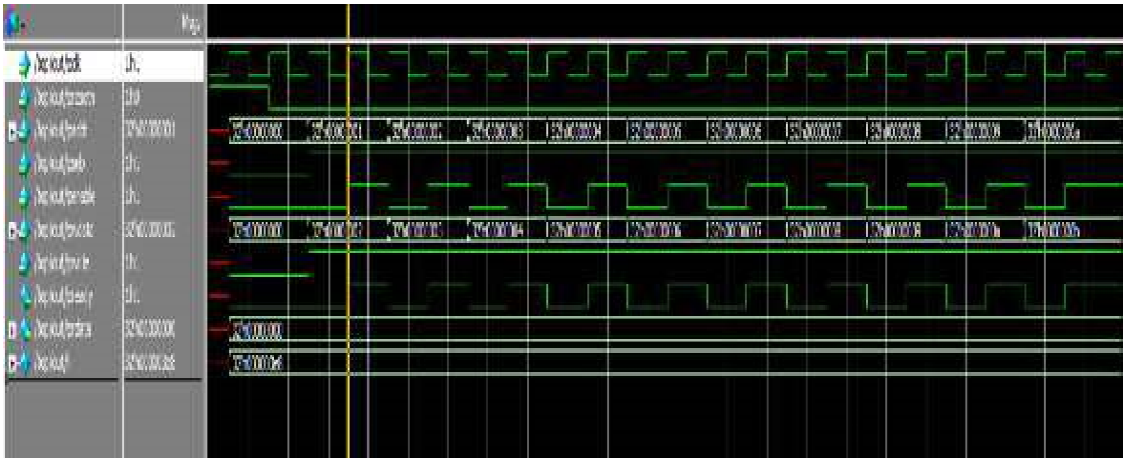

Figure 8. Multiple write transfer without wait

In this above figure 8, each write transaction will take 2 clock cycle and PENABLE will not wait for PREADY signal. Automatically both the PENABLE and PREADY are asserted high in same clock pulse.

4.3. Multiple write and read transfer.

Figure 9. Multiplewrite and read transfer without wait

4.4 Output for single with write wait statement.

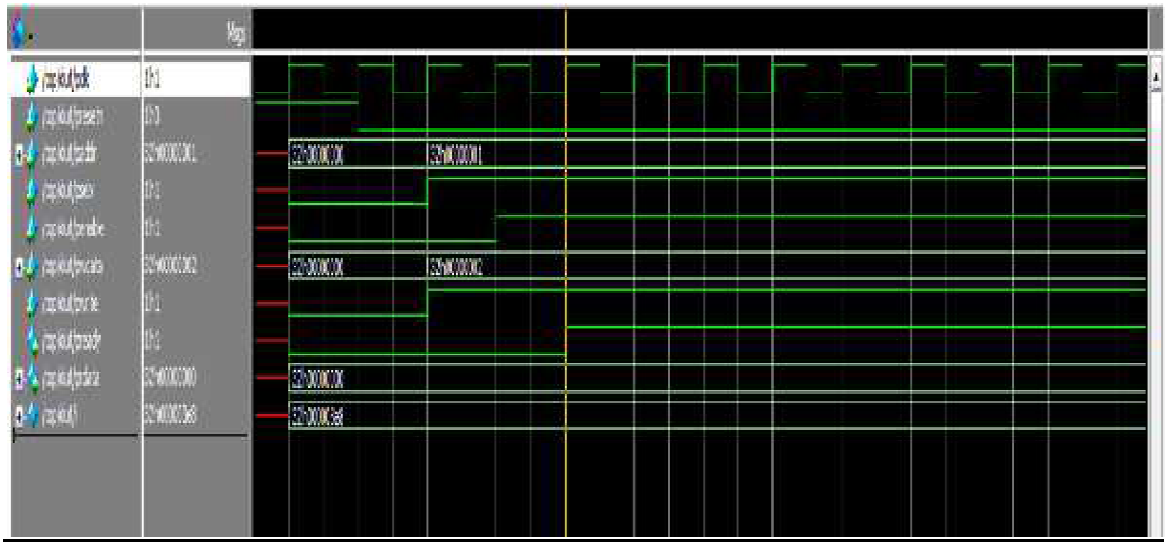

Figure 10. Single write with wait statement

4.5 Output for multiple write with wait statement.

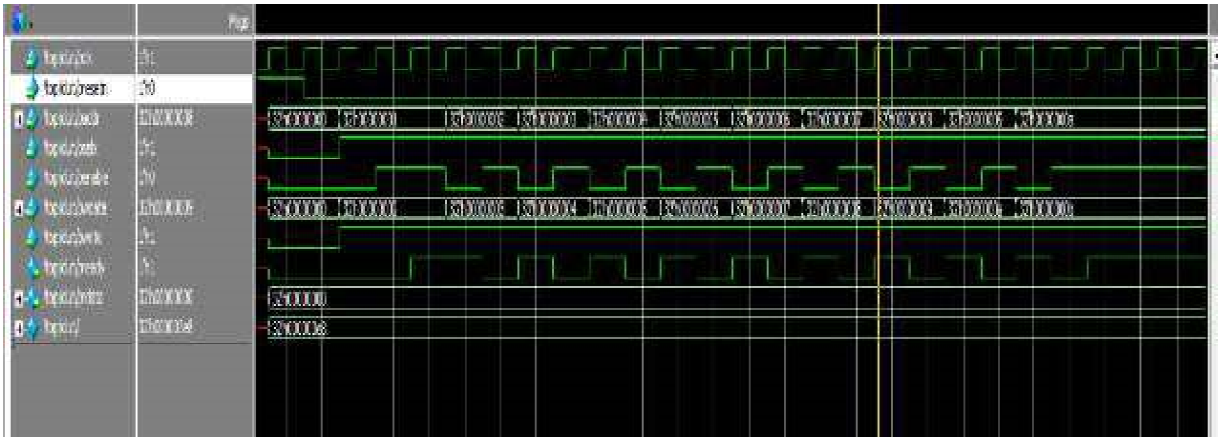

Figure 11. Multiple write with wait

In above mentioned figure 11. First transaction alone takes 3 clock cycles because PENABLE remains high until PREADY also asserted high. And from in next transaction it will take 2 clock cycles to complete each write transactions.

4.6. Output for multiple read and write with wait statement.

In multiple write and read transaction, write transaction completed first and followed by that read transaction takes place.

During first write transaction PENABLE signal waits until PREADY gets asserted high (1) so here three clock cycles is required. And the following write transaction APB will works on two clock cycles (figure 12).
While in multiple read transfer we can't notice PENABLE wait for PREADY. Because wait statement is designed only for the first clock cycle when PRESETn asserted to zero [5].

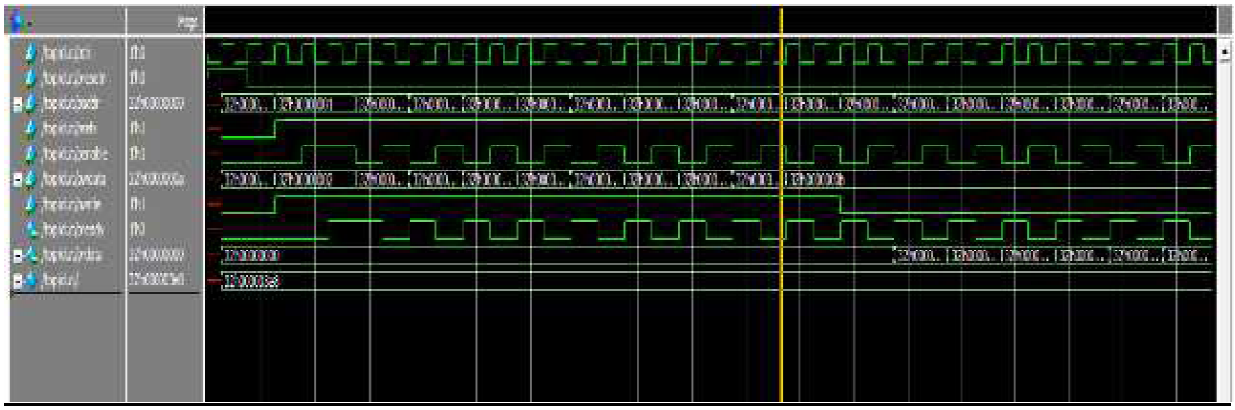

Figure 12. Multiple read and write with wait statement.

5. CONCLUSION

This paper provides in detail the outline of the AMBA bus architecture and APB protocol in detail. According to the above mentioned specification, APB is designed and verified using QUESTASIM. Here we summarized the six test cases, single and multiple write transaction with and without wait, multiple write transaction with and without wait states, multiple read and write transaction with and without wait. Hence the system is functionally correct. QUESTASIM also ensures the functional correctness of the design.

References

[1] ARM, "AMBA Specification Overview", http://www.arm.com/. .
[2]APB Example-AMBA system, Technical reference manual,ARM Inc.,1999.
[3] Akhilesh Kumar, Richa Sinha, "Design and Verification analysis of APB3 Protocol with Coverage,"IJAET, Nov 2011.
[4] SanthiPriyaSarekokku, K. Rajasekhar, "Design and Implementation of APB Bridge based on AMBAAXI 4.0," IJERT, Vol.1, Issue 9, Nov 2012.
[5] VERILOG Reference Manual, http://www.accellera.com
[6] Samir Palnitkar, "Verilog HDL: A guide to Digital Design and Synthesis (2nd Edition), Pearson, 2008.