Design and Verification of Bridge between APB (AMBA) and Avalon Bus

Rakesh Reddy Balpunuri[1], Venkat Rao P[2]

Abstract – To establish a communication between two different devices that are running on different frequencies with distinct protocols, we have to design a BRIDGE. This BRIDGE that acts as an interface to the both modules synchronizes their signals. The main aim is to design and thoroughly verify the bridge between two Protocols that are majorly used in the Real Time World. The protocols used here are APB (Advanced Peripheral Bus) defined in AMBA (Advanced Microcontroller Bus Architecture) specifications developed by ARM, and AVALON Protocol developed by ALTERA.

Index Terms – AMBA APB, Avalon, Bridge, Questasim, System Verilog, Verilog.

1 INTRODUCTION

As the technology is growing day by day, it has become a challenge for digital designer to develop optimized designs that suits the customer requirements. There is a heavy competition for designing and marketing of smart electronic products or smart electronic gadgets that decides the fate of the companies.

Internal to every smart design, there could be many modules that are used to attain the required functionality. Each module may run with different frequencies. Every module has their own module level signals, operating voltages and operating frequencies. In order to build a complete smart device all these modules with different characteristics are to be interconnected. For this purpose there are certain protocols in the market that can be used to achieve an efficient performance of the device. To establish a communication between two different devices that are running on different frequencies with distinct protocols, we have to design a BRIDGE. This BRIDGE that acts as an interface to the both modules synchronizes their signals.

In this project the main aim is to design and thoroughly verify the bridge between two Protocols that are majorly used in the Real Time World. The protocols used here are APB (Advanced Peripheral Bus) defined in AMBA[1] (Advanced Microcontroller Bus Architecture) specifications developed by ARM, and AVALON[2] Protocol developed by ALTERA.

[1]*Master of Technology in VLSI & Embedded Systems. DRK College of Engineering & Technology. PH-+91-9030596932.*
E-mail:rakesh.jot@gmail.com.

[2]*Head of Department of Electronics & Communication Engineering. DRK College of Engineering &Technology.*

2 Basic Overview of APB & Avalon Protocol

2.1 APB Protocol Overview

AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus. The APB interfaces to any peripherals that are low-bandwidth and do not require the high performance of a pipelined bus interface. The APB has unpipelined protocol. All signal transitions are only related to the rising edge of the clock to enable the integration of APB peripherals easily into any design flow. Every transfer takes at least two cycles.

2.3 Avalon Protocol Overview

The Avalon bus is a simple bus architecture designed for connecting on-chip processors and peripherals together into a system–on–a–programmable chip (SOPC). The Avalon bus is an interface that specifies the port connections between master and slave components, and specifies the timing by which these components communicate. The Avalon bus module is the backbone of a system module. It is the main path of communication between peripherals components in an SOPC design. The Avalon bus module is the sum of all control, data and address signals and arbitration logic that connect together the peripheral components making up the system module. The Avalon bus module implements a configurable bus architecture, which changes to fit the interconnection needs of the designer's peripherals.

3 Block Diagram of APB to Avalon Bridge

Figure 1 shows the different blocks in top level reference design. The block that is present on the left-hand side is the APB bus which acts as Master to the Bridge. The block diagram that is present on right-hand side is the Avalon bus that acts as slave to the bridge.

The block that is present in the middle is the main design (Bridge) that synchronizes the APB bus signals to that of Avalon bus signals. This block contains the complete control logic based on which the communications (Transactions) between both the devices occurs.

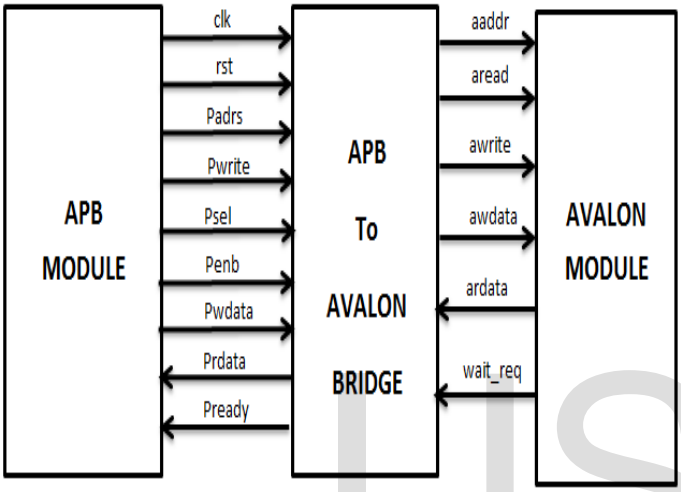

Figure 1 Top Level Block Diagram

4 AMBA APB Bus Protocol

4.1 APB Operating States

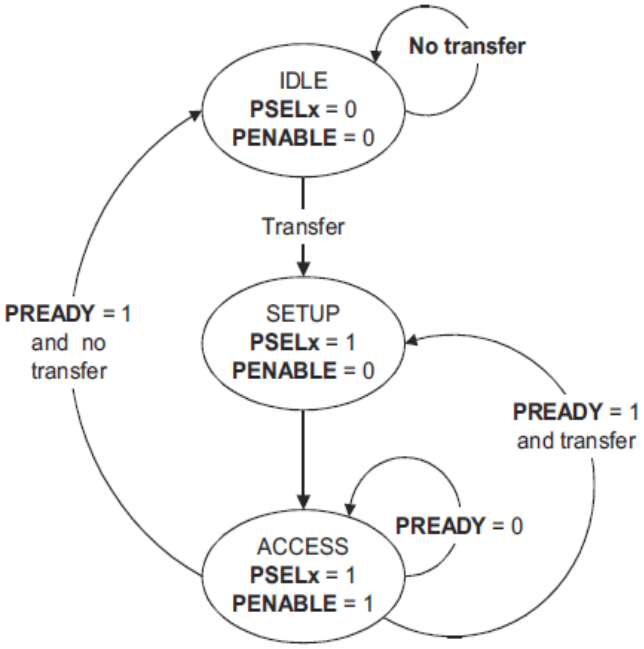

Figure 2 State Diagram

The state machine operates through the following states:

- **IDLE** This is the default state of the APB.
- **SETUP** When a transfer is required the bus moves into the SETUP state, where the appropriate select signal, **PSELx**, is asserted. The bus only remains in the SETUP state for one clock cycle and always moves to the ACCESS state on the next rising edge of the clock.
- **ACCESS** The enable signal, **PENABLE**, is asserted in the ACCESS state. The address, writes, select, and write data signals must remain stable during the transition from the SETUP to ACCESS state. Exit from the ACCESS state is controlled by the **PREADY** signal from the slave:
  - If **PREADY** is held LOW by the slave then the peripheral bus remains in the ACCESS state.
  - If **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. Alternatively, the bus moves directly to the SETUP state if another transfer follows.

4.2 APB Bus Signal Description

signal	Description
pclk	Clock. The rising edge of **PCLK** times all transfers on the APB
prstn	Reset. The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal
paddr	Address. This is the APB address bus. It can be up to 32 bits wide.
psel	Select.. It indicates that the slave device is selected and that a data transfer is required. There is a Pselx signal for each slave.
Penb	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
pwrite	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW
pwdata	Write data. This bus is driven by the Master during write cycles when **PWRITE** is HIGH. This bus can be up to 32 bits wide.
Pready	Ready. The slave uses this signal to extend an APB transfer.
Prdata	Read Data. Driven by slave when **PWRITE** is LOW. This bus can be up to 32-bits wide.

4.2 APB Write Transfers

The write transfer starts with the address, write data, write signal and select signal all changing after the rising edge of the clock. The first clock cycle of the transfer is called the Setup phase. After the following clock edge the enable signal is asserted, **PENABLE**, and this indicates that the Access phase is taking place. The address, data and control

signals all remain valid throughout the Access phase. The transfer completes at the end of this cycle. The enable signal, **PENABLE**, is deasserted at the end of the transfer. The select signal, **PSELx**, also goes LOW unless the transfer is to be followed immediately by another transfer to the same peripheral. Figure 3 below shows the write transactions.

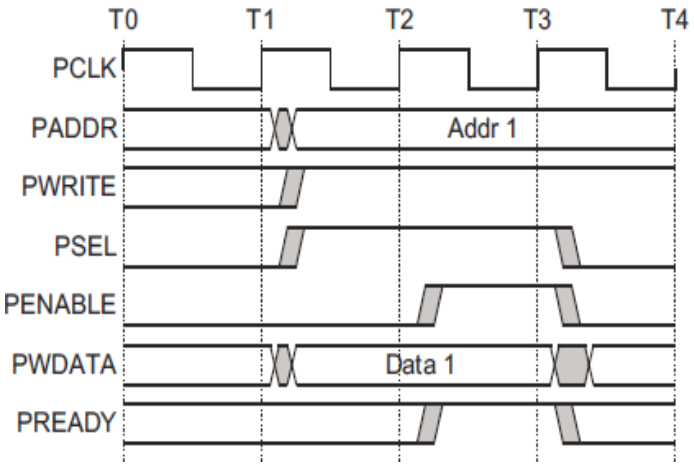

Figure 3 APB Write Transfer waveforms

4.3 APB Read Transfer

Figure 4 shows a Read Transfer. It starts with the address, read signal and select signal all changing after the rising edge of the clock. The first clock cycle of the transfer is called the Setup phase. After the following clock edge the enable signal is asserted, **PENABLE**, and this indicates that the Access phase is taking place. The address, data and control signals all remain valid throughout the Access phase. The transfer completes at the end of this cycle. The enable signal, **PENABLE**, is deasserted at the end of the transfer. The slave must provide the data before the end of the read transfer.

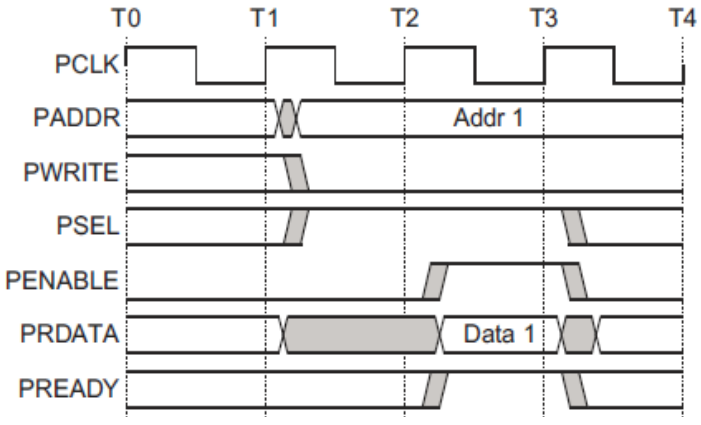

Figure 4 APB Read Transfer waveforms

5 AVALON Bus Protocol

5.1 Avalon Bus Signal Description

Signals	Descriptions
Aaddr	Address lines to Avalon Module
Aread	Read Request Signal to Slave(Avalon module)
Awrite	Write Request Signals to Slave(Avalon Module)
Awdata	Data line to Avalon Module for Write Transfer.
Ardata	Data lines from Avalon Module for Read Transfer
Wait_req	Used to stall the transactions when salve (Avalon module) is busy and not able to respond

5.2 Avalon Write Transfers

The master write transfer starts on the rising edge of clk. immediately after the first rising edge of clk, the master asserts the address, writedata and write signals. If the data cannot be captured by the next rising clock edge, the Avalon bus module asserts wait request during the first bus cycle. The master must keep address, writedata and write asserted constantly until the next rising clock edge after wait request is deasserted. After wait request is deasserted, the master port deasserts address, read data and read on the next rising edge of clk. The master may initiate another master transfer during the next bus cycle.

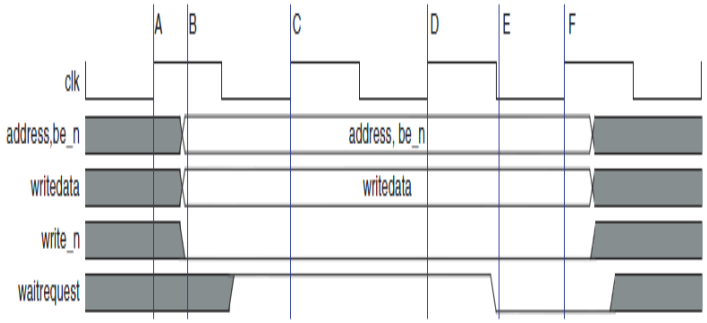

Figure 5 Avalon write transfer waveforms

5.3 Avalon Read Transfers

The master read transfer starts on the rising edge of clk. immediately after the first rising edge of clk, the master asserts the address and read signals. If the Avalon bus module cannot present read data within the first bus cycle, it asserts wait request before the next rising edge of clk. If the master sees wait request asserted on the rising edge of clk, then it waits. The master must hold all outputs constant until the next rising clock edge after wait request is

deasserted. After wait request is deasserted, the master port then captures read data on the next rising edge of clk, and deasserts address and read. The master may initiate another transfer immediately during the next bus cycle.

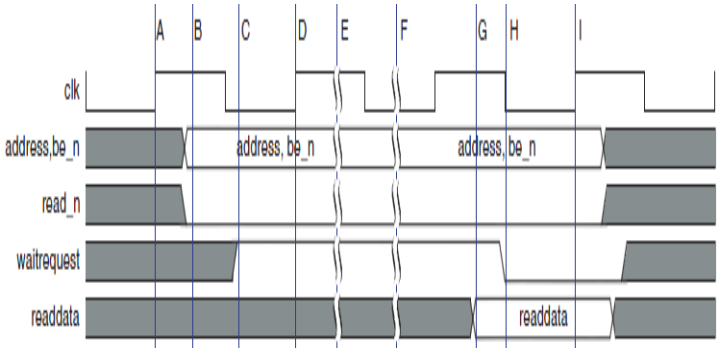

Figure 6 Avalon Read Transfers

6 Design & Verification Environment

6.1 Design of RTL

The Bridge that acts as a connecting interface between APB Module and Avalon Module is designed as an RTL (Register Transfer Level). Verilog language is used to design this RTL design.

6.2 Verification Environment

Any design is said to be completely done only if it is thoroughly verified with all the different scenarios including corners cases. This verification can be done in many ways. Verilog or System Verilog can be used to develop a verification environment.

As said above, the term Verification Environment comprises of many different modules that are used to verify the Design Under Test (DUT). They can listed as follows-

1. Driver – Drives the DUT
2. Monitor – Captures output of DUT and sends packets to Scoreboard.
3. Scoreboard – Compares data obtained from Monitor with that of Expected data.
4. Test bench – Acts as an interface between Verification Environment and DUT.

6.3 Simulations

Simulations are carried out using Questasim[3] tool developed by Mentor Graphics using either Verilog or System Verilog as programming language. The test case is run for multiple operations and the waveforms are visible in GUI mode.

7 RESULTS

Simulations are done using Questasim tool in command window with Verilog as design language and System Verilog as Verification Language.

Compilation and Simulation can be triggered either in command mode or GUI mode. Screen shots of compilation and simulations can be seen as follows.

Figure 7.1 Shows the RTL Data flow of Bridge between APB & Avalon.

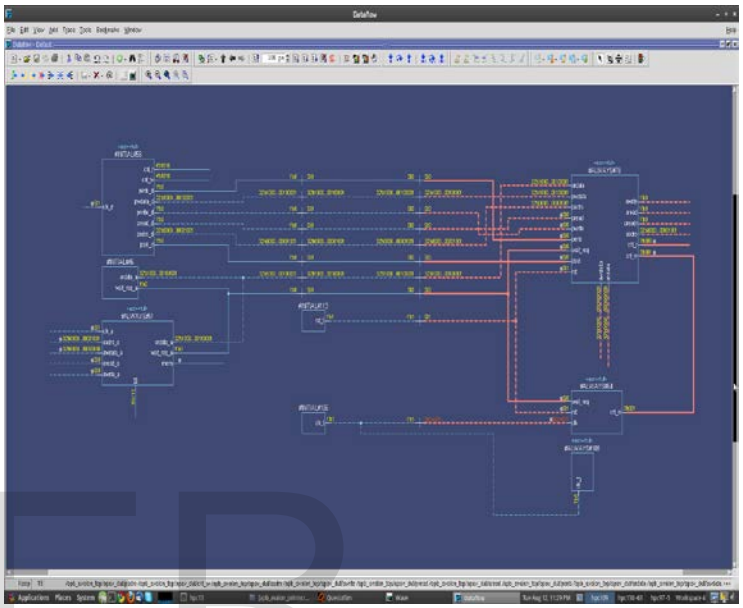

Figure 7.1 RTL Dataflow Schematic

Figure 7.2 Shows Schematic Dataflow of Top-level.

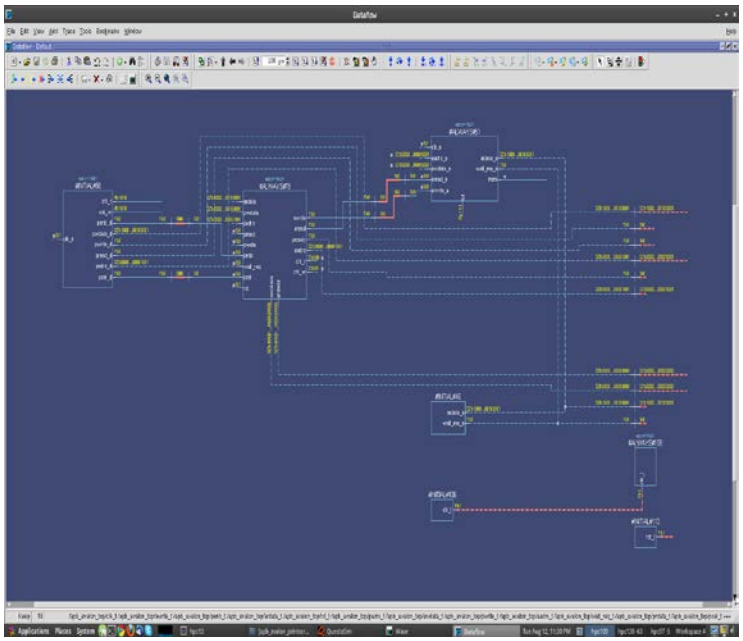

Figure 7.2 Top-level Schematic

Figure 7.3 Shows Simulation Waveform for Write Transfers.

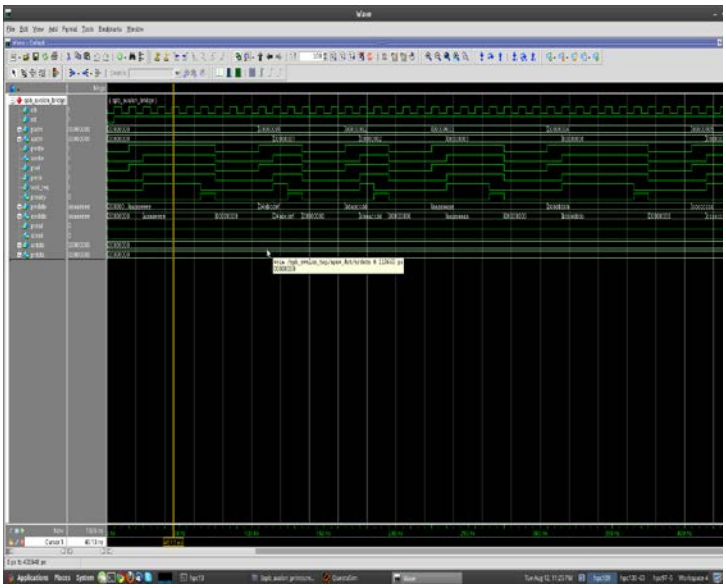

Figure 7.3 Waveforms for Write Transfers

Figure 7.4 Shows Simulation Waveform for Read Transfers

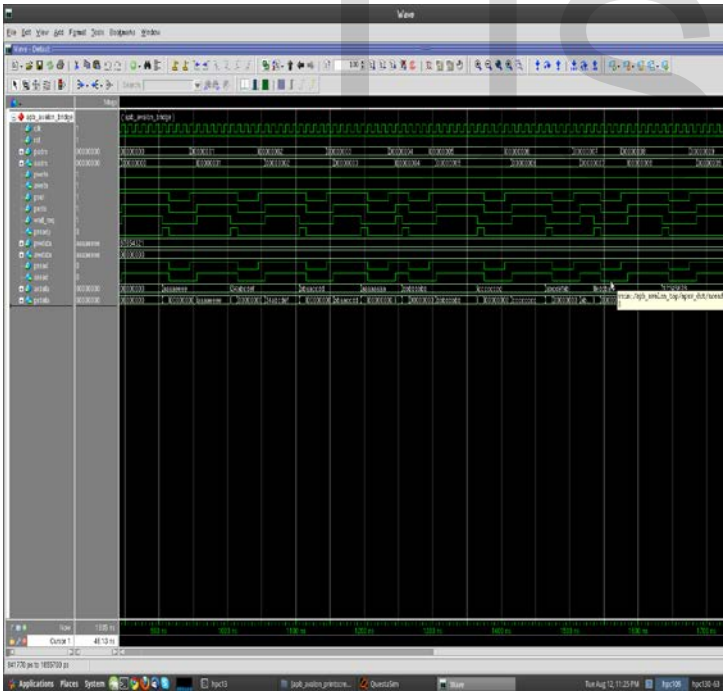

Figure 7.4 Waveforms for Read Transfers

8 Conclusion

AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus. The Avalon bus is a simple bus architecture designed for connecting on-chip processors and peripherals together. In this work APB acts as Master whereas Avalon acts as Slave. Thus, bridge between APB bus and Avalon is designed and implemented successfully.

REFERENCES

[1] AMBA 3 APB Protocol Specification (Rev 1.0) available at http://www.arm.com. 17th August 2004

[2] Avalon Bus Specification Reference Manual available at http://www.altera.com. Rev 2.1, January 2003

[3] Questa® SIM User's Manual with Software Version 10.0c is proprietary to Mentor Graphics Corporation available at www.mentor.com.

[4] Verilog language basics at http://www.asic-world.com/verilog/index.html.

[5] System Verilog language for Verification Environment - http://www.asic-world.com/systemverilog/tutorial.html.