ADVANCE PERIPHERAL BUS (APB)

VERSION 1.0

Revision History

Revision	Description	Date	Modified By
1.0	Initial Document	13/05/2024	
1.1	--	--	--

Table of Contents

Chapter 1: APB Overview

- The APB is part of the AMBA 3 protocol family. It provides a low-cost interface that is optimized for minimal power consumption and reduced interface complexity.
- 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.
- The APB can interface with the AMBA Advanced High-performance Bus Lite (AHB-Lite) and AMBA Advanced Extensible Interface (AXI).

Chapter 2: APB features

The key features of the APB are:

1. Write/Read with no wait state
2. Write/Read with wait state
3. Error Response
4. Minimal power consumption
5. Unpipelined protocol

Chapter 3: APB Verification Plan

3.1 Feature Extraction

1. Normal Scenario: This will check basic read and write operation.
2. Write with no wait state: This test will write data into slave without wait as PREADY gets asserted with PENABLE.
3. Write with wait state: This test will write data into slave with wait state.
4. Read with no wait state: This test will read the data from slave without waiting.
5. Read with wait state: This test will read the data from slave with wait state.
6. Reset on fly: to check reset in between.
7. Error response: to check the PSLVERR is getting asserted when error in signals.

3.2 Coverage Plan

Cover different scenarios of operation, including:

1. Normal Operation: Transactions under normal operating conditions.
2. Edge Cases: Transactions at boundary conditions or extreme values.
3. Error Handling: Transactions involving error conditions and fault recovery.

3.3 Checker Plan

In-order scoreboard: Ensure that the data sent by the master is correctly received by the APB slave during write transactions. Verify that the data read by the master matches the expected data from the APB slave during read transactions.

Reset Checker: Checks that when reset is asserted, all the signal becomes zero or not.

Slave Error checker: If an operation is performed at an inaccessible address then error signal PSLVERR should be asserted by the slave. So to compare expected and actual PSLVERR signal to verify the error, this checker was implemented.

3.4 Verification Environment Development

3.5 Test suite development

3.5.1 Directed Testcases

3.5.2 Random Testcases

Chapter 4: APB Verification Environment Development

4.1 APB Block Diagram

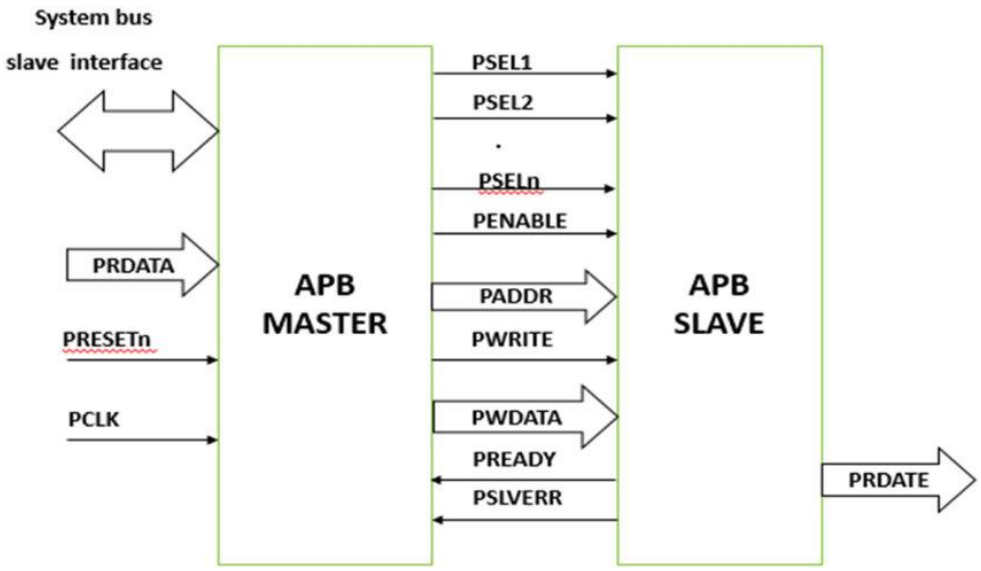

Figure 4.1: APB Block Diagram

4.2 Verification Architecture

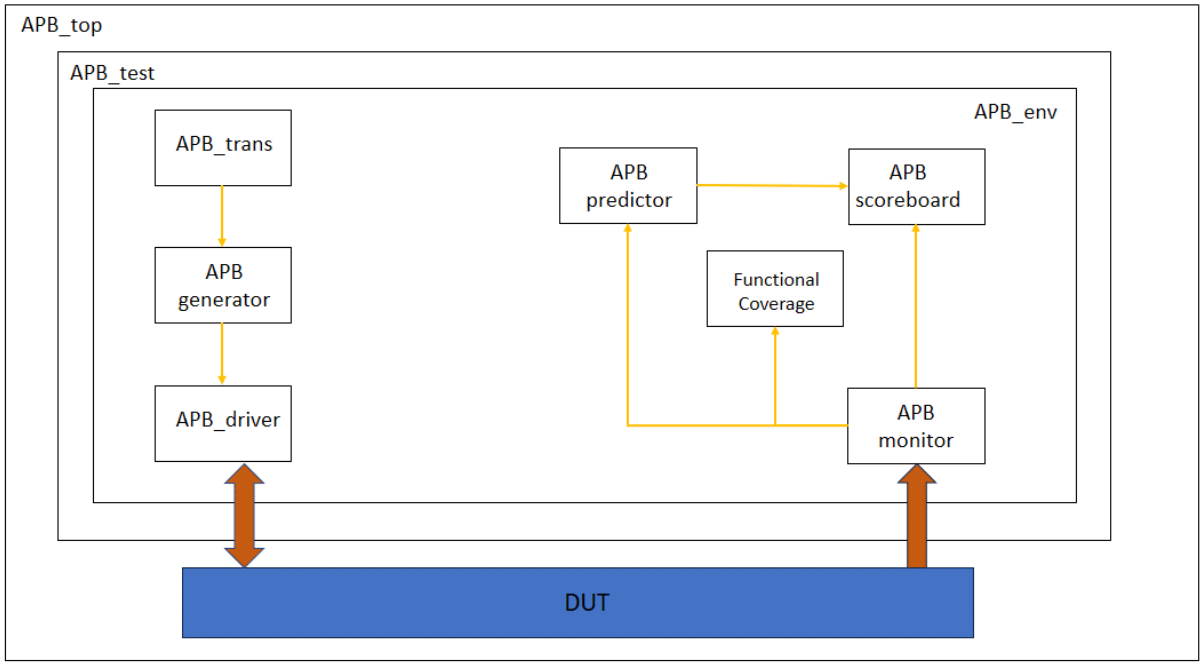

Figure 4.2: APB Verification Architecture in System Verilog

4.3 APB Components

4.3.1 Transaction Class

In transaction class, signals required to be randomized will be declared as of “rand” type, output signals from the slave will be included. The packet (object) of transaction class be available in every component through which data will be passed using mailbox between the components. It also contains an enum type variable which have operations to be performed like WRITE, READ or RESET. Enum will also be of “rand” type. Following are the signals included in the transaction class.

- PADDR (rand type)
- PWDATA (rand type)
- PSEL
- PENABLE
- PWRITE
- PRDATA
- PSEL
- PREADY
- PSLVERR
- Ops_e (enum type Variable of rand type)

4.3.2 Generator Class

In generator class, main logic to implement test cases will be written. Here, an abstract generator class is declared where all the required variables and methods will be declared. Then this abstract class will be extended and logic for run_phase method will be added as per the testcases requirements.

4.3.3 Driver Class

Here, transaction level values will be taken from generator class using mailbox and will be driven as per the protocol to the interface. Here, transaction level data will be converted to pin level data. When all the values are passed to interface, driver will wait for the next packet to arrive in the mailbox and further drive the values.

4.3.4 Monitor Class

The monitor observes the signals on the APB bus and captures transaction-level information. It extracts relevant details from the APB protocol signals, such as addresses, data, control signals, and response signals, and forwards this information to the scoreboard for verification.

4.3.5 Reference Model

The checker module performs detailed protocol and functional checks on the transactions observed by the monitor. It verifies that the APB slave adheres to the APB protocol specifications and operates correctly under various scenarios. Additionally, the checker ensures that the DUT behaves as expected and raises alerts for any violations or errors encountered during verification.

4.3.6 Scoreboard

The scoreboard compares the expected results with the actual results obtained from the DUT. It checks for correctness, completeness, and timing of transactions by comparing the transactions generated by the driver with the observed transactions from the monitor. Any discrepancies are flagged and reported for further analysis.

Chapter 5: Running Simulation

Chapter 6: Closure Report