**VERIFICATION PLAN**

Verification plan for an AHB-to-APB Bridge involves defining a systematic approach to verify that the bridge meets its functional, performance, and design specifications. Below is a detailed verification plan outline for the AHB-to-APB Bridge:

**1. Objective**

To verify the functionality, performance, and compliance of the AHB-to-APB Bridge as per the design specifications and protocol standards.

**2. Features to Verify**

* **AHB Interface:**
  + Support for all AHB protocol transactions (READ/WRITE, BURST types, single transfers, etc.).
  + Proper handling of AHB signals such as HREADY, HRESP, HWRITE, HADDR, HSIZE, and HTRANS.
  + Pipeline and non-pipeline operation.
* **APB Interface:**
  + Generation of correct APB transactions (PWRITE, PSEL, PENABLE, PADDR, PWDATA).
  + Idle, setup, and access phases on APB.
* **Bridge-Specific Functionalities:**
  + AHB to APB clock domain crossing, if applicable.
  + Correct address mapping between AHB and APB.
  + Handling of various HSIZE and alignment constraints.
  + Proper handling of bus contention and error responses.

**3. Verification Methodology**

* **Verification Approach:**
  + **Directed Tests:** To validate individual features or scenarios.
  + **Randomized Testing:** To uncover corner cases by randomizing inputs and transaction sequences.
  + **Coverage-Driven Verification:** To ensure all features and scenarios are exercised.
  + **Assertions and Monitors:** To verify protocol compliance and detect incorrect behaviour.
* **Testbench Architecture:**
  + Use of Universal Verification Methodology (UVM) for modularity and reusability.
  + Components:
    - AHB Driver: Generates AHB transactions.
    - AHB Monitor: Observes and checks AHB transactions.
    - APB Driver: Generates responses for APB transactions.
    - APB Monitor: Observes and checks APB transactions.
    - Scoreboard: Compares expected vs. actual transactions.
    - Coverage Collector: Measures functional and code coverage.
    - Environment: Integrates the drivers, monitors, scoreboard, and DUT.

**4. Test Scenarios**

**Basic Functional Tests:**

* Simple AHB to APB transactions (single READ/WRITE).
* Different transfer sizes (byte, half-word, word).
* Single and burst transfers.

**Protocol Compliance Tests:**

* AHB protocol timing checks (valid HREADY, HRESP handling).
* APB setup and hold timing.
* Handling of wait states and errors.

**Stress and Randomization Tests:**

* Randomized AHB transactions with varying sizes, addresses, and burst types.
* Concurrent read and write operations.

**Corner Cases:**

* Unaligned address handling.
* Burst wrap-around addressing.
* Back-to-back transactions with no idle cycles.
* Clock domain crossing (if applicable).

**Error Handling:**

* Slave errors.
* Invalid transactions on AHB or APB.
* Data mismatch or corrupted data transfers.

**Performance Testing:**

* Throughput and latency measurements.
* Maximum clock frequency validation

**5. Functional Coverage**

Define coverage points for:

* All AHB transfer types (SINGLE, INCR, WRAP).
* All HSIZE values.
* All burst lengths and wrap scenarios.
* Different APB access patterns (WRITE/READ).
* Error scenarios and protocol violations.

**6. Assertions**

Write assertions for:

* Protocol rules for AHB and APB.
* Timing constraints between signals.
* Correct transfer of data and control signals between AHB and APB.

**7. Tools and Environment**

* **Simulation Tools:** VCS, QuestaSim.
* **Coverage Tools:** Functional coverage using tools like Synopsys VCS.
* **Waveform Viewer:** For debugging and analysis.

**8. Metrics for Completion**

* 100% functional coverage for all defined features and scenarios.
* High code coverage (line, toggle, FSM).
* All directed and randomized tests passing.
* Assertions with no failures.

**9. Deliverables**

* Testbench code and verification environment.
* Test case results and reports.
* Functional coverage and code coverage reports.
* Bug reports and resolution details.