**Project Proposal: Functional Verification of an AMBA AXI4-Lite Protocol Using SystemVerilog UVM**

**1. Title**

**"Functional Verification of AXI4-Lite Protocol Using SystemVerilog UVM: A Reusable Testbench Approach"**

**2. Introduction**

The **AMBA AXI4-Lite protocol** is widely used for on-chip communication in SoCs due to its simplicity and high performance. Verifying such protocols manually is time-consuming and error-prone. The **Universal Verification Methodology (UVM)** provides a reusable and scalable way to verify digital IPs.

This project focuses on **developing a UVM-based testbench** to verify an **AXI4-Lite slave interface**, covering basic read/write operations and protocol checks, suitable for beginners in digital verification.

**3. Objectives**

* Understand the AXI4-Lite protocol.
* Develop a synthesizable AXI4-Lite slave model (or use an open-source one).
* Create a reusable **UVM testbench**.
* Verify read/write operations and error scenarios.
* Perform **functional coverage** and assertion-based checks.
* Generate a basic report suitable for publication.

**4. Scope of Work**

1. **Design/Use AXI4-Lite Slave IP**
   * Use a simple RTL design with AXI4-Lite slave interface.
   * Optionally create a simple register block (for read/write).
2. **Testbench Development**
   * Create UVM components: agent, driver, monitor, sequencer, scoreboard.
   * Write basic sequences: read, write, invalid address, burst errors.
3. **Functional Verification**
   * Simulate using **ModelSim**, **Questa**, or **Vivado Simulator**.
   * Include SystemVerilog assertions (SVAs) for protocol checking.
   * Implement functional coverage (optional).
4. **Result Analysis**
   * Pass/fail logs.
   * Coverage report.
   * Debugging with waveform viewer.

**5. Tools and Languages**

* **Languages**: SystemVerilog (for UVM), Verilog (for RTL)
* **Tools**: ModelSim/QuestaSim, Vivado (optional), Git
* **Methodology**: UVM (Universal Verification Methodology)

**6. Expected Outcomes**

* Fully functional AXI4-Lite testbench in UVM.
* Verified RTL design with 100% functional coverage (basic cases).
* Waveform and log analysis.
* Clean, modular testbench for reuse.
* A short conference paper describing:
  + Verification plan
  + UVM architecture
  + Simulation results

**7. Project Timeline (10–12 weeks)**

| **Week** | **Task** |
| --- | --- |
| 1–2 | Study AXI4-Lite protocol & UVM basics |
| 3–4 | AXI4-Lite RTL design (or reuse one) |
| 5–6 | Develop UVM components |
| 7–8 | Write test sequences & run simulations |
| 9 | Add assertions and coverage |
| 10 | Analyze results and write report/paper |

**8. Suitability for IEEE Paper**

This project is suitable for short papers at:

* **IEEE VLSI Test Symposium (VTS)**
* **IEEE International Symposium on Quality Electronic Design (ISQED)**
* **IEEE International Conference on VLSI Design and Embedded Systems**

Topics could include:

* "Teaching UVM through Simple Protocol Verification"
* "Reusable Testbench for On-Chip Protocols"

**9. References**

1. ARM AMBA AXI Protocol Specification (Lite subset)
2. *SystemVerilog for Verification*, Chris Spear
3. UVM 1.2 User Guide (Accellera)
4. IEEE Standard 1800-2017: SystemVerilog