🔧 Mini Project Challenge – Phase 2

AXI-Lite Packet Validator & Sorter

Theme: *Data Validation* | *Packet Sorting* | *AXI* Fundamentals | Verilog | VHDL | SV

🔧 Objective:

Design an **AXI4-Lite Slave Peripheral** that can **receive data packets**, **validate** them based on a rule, and **sort** them into valid or invalid storage.

You'll learn to handle **AXI write transactions**, implement **basic control logic**, and design **simple FIFO-like sorting** — key building blocks in many real-world FPGA systems.

📦 System Overview:

1. **Write Address 0x00**:
   Send a 32-bit packet.
2. **Write Address 0x04**:
   Acts as a *"commit"* signal — triggers the validation and sorting of the last packet.
3. **Validation Rule:**
   A packet is considered **valid** if bits [31:24] equal 8'hA5.
   Otherwise, it is **invalid**.
4. **Storage:**
   - Valid packets are stored in a valid_fifo[0:7]
   - Invalid packets go to invalid_fifo[0:7]

📐 AXI-Lite Interface Requirements:

- Implement a minimal AXI4-Lite write interface:
  - AWVALID, AWREADY, AWADDR
  - WVALID, WREADY, WDATA

    - BVALID, BREADY, BRESP

No read interface is required for this challenge.

🧪 **Expected Testbench Behavior:**

- Send at least 5 packets:
    - Mix of valid and invalid ones.
- After each packet write to 0x00, write any value to 0x04 to commit it.
- Confirm packets are sorted into the correct FIFOs.

Google forms will be shared soon for uploading your responses.