Overview of Router 1×3 Verification Using UVM

1. Introduction

- A **1×3 Router** is a digital circuit that receives data from a **single input** and routes it to one of **three output ports** based on a **control signal**. It ensures efficient data transmission in communication systems by directing packets to their intended destinations.
- To guarantee correct operation, the router undergoes **functional verification** using **Universal Verification Methodology (UVM)**. The goal is to validate that the router correctly processes incoming data, avoids packet loss, and handles different traffic conditions effectively.

---

2. Router 1×3 RTL Design

Key Features of the Router

- **Single input, three output ports**
- **Control logic** to decide the routing path
- **Data buffering** for managing congestion
- **Handshake signals** for flow control

Important Signals

Signal Name	Description
clk	System clock
reset	Active-low reset signal
data_in	Input data packet
addr	Destination address (0,1,2)
valid_in	Indicates a valid packet
ready_out[3]	Readiness status of each output port
data_out[3]	Routed output data

---

3. Verification Strategy Using UVM

Goals of Verification

- Ensure correct routing of data packets.
- Check system behavior under normal and extreme conditions.
- Validate performance under **randomized traffic conditions**.
- Achieve high **functional and code coverage**.

Key Test Scenarios

1. **Basic Functionality Test** – Verify correct routing based on addr.
2. **Back-to-Back Transfers** – Check continuous data flow without errors.
3. **Output Busy Condition** – Ensure proper handling when an output is full.
4. **Randomized Testing** – Generate random packets to simulate real-world behavior.
5. **Corner Cases** – Test edge conditions like reset behavior and extreme loads.

---

4. UVM Testbench Architecture

The UVM testbench follows a modular approach with the following components:

(i) Interface (router_if.sv)

- Defines the **signals and handshake logic** between the testbench and DUT.

(ii) Testbench Top (top.sv)

- Instantiates the **Router 1×3 RTL** and the UVM **testbench environment**.

(iii) UVM Environment (env.sv)

- **Driver**: Sends test packets to the DUT.
- **Monitor**: Observes transactions at inputs and outputs.
- **Agent**: Groups Driver and Monitor for modular design.
- **Scoreboard**: Compares expected vs. actual results.
- **Test**: Defines scenarios and controls the verification flow.

---

5. Coverage Metrics & Debugging

To ensure complete verification, both **functional coverage** and **code coverage** are analyzed:

- **Functional Coverage**:
    - Coverage points for all possible addresses (addr = 0,1,2).
    - Verification of busy and non-busy conditions.
    - Different packet sizes and burst transfers.
- **Code Coverage**:
    - **Statement coverage** – Ensures all code lines are executed.
    - **Branch coverage** – Validates if all control paths are tested.
    - **Toggle coverage** – Ensures all signal transitions are checked.

---

6. Conclusion

The **Router 1×3 verification** ensures the correct functionality, reliability, and robustness of the router. By using **UVM**, a structured and reusable test environment is built to simulate real-world data traffic, identify issues, and validate the design under different scenarios.