



Digital Design Verification

Lab Manual #42 – TLM Connections

Release: 1.0

**Date: 18-Aug-
2024**

NUST Chip Design Centre (NCDC), Islamabad, Pakistan



Copyrights ©, NUST Chip Design Centre (NCDC). All Rights Reserved. This document is prepared by NCDC and is for intended recipients only. It is not allowed to copy, modify, distribute or share, in part or full, without the consent of NCDC officials.

Revision History

Revision Number	Revision Date	Revision By	Nature of Revision	Approved By
1.0	18/08/2024	Amber Khan	Complete Manual	-



Contents

Contents.....	2
Objective	3
Tools	3
Instructions for Lab Tasks	3
Task 1: Using TLM Export Connectors.....	3
Task 2: Using TLM Analysis FIFOs	4



Objective

The objective of this lab is

- To use export TLM connections with the router module UVC.
- To build a scoreboard using TLM Analysis FIFOs.

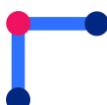
Tools

- SystemVerilog
- Cadence Xcelium

Instructions for Lab Tasks

The submission must follow the hierarchy below, with the folder named after the student (no spaces), and the file names exactly as listed below.

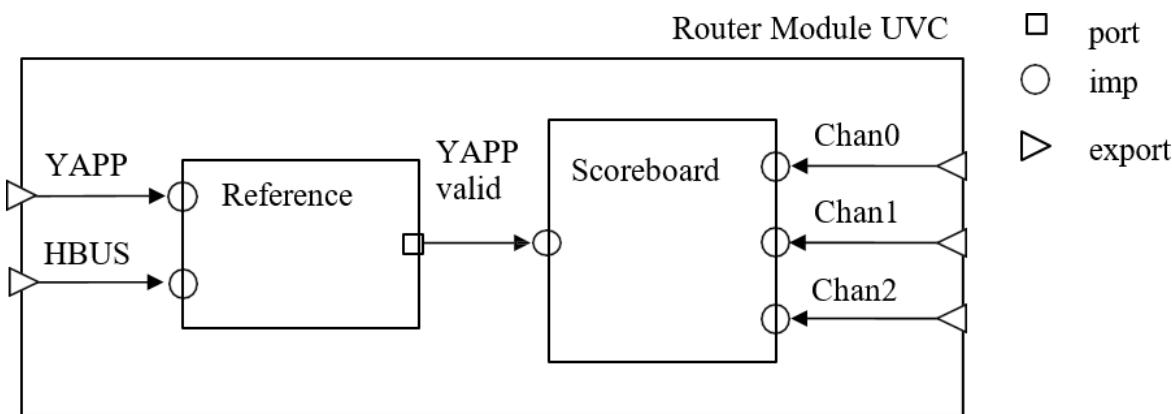
```
./student_name_lab14/
    └── task1/
        ├── sv/
        └── tb/
    └── task2/
        ├── sv/
        └── tb/
    └── router_rtl
    ├── yapp
    ├── hbus
    ├── channel
    └── router_module_uvc
    └── clk_and_reset
```



Task 1: Using TLM Export Connectors

A module UVC does not have such a clearly defined architecture as an interface UVC. With an interface UVC, TLM analysis **port** objects are always defined in the monitor components. We know where to look in order to find these. With a module UVC, it can be more difficult to find declarations for the TLM analysis **imp** objects.

One technique is to extract all the TLM objects used in the module UVC to the top level environment. These top-level objects are then connected internally to the correct sub-component.



1. Modify your router module UVC to present the external TLM connections as top-level objects of the router module **env**. Note the following:
 - You will need to declare **exportobjects** in the Router Module environment for the **imp** objects of the monitor and scoreboard.
 - You must connect all the **exportobjects** to **impobjects** in the Router Module **connect** method.
 - The internal YAPP valid connection between reference and scoreboard does not need to be routed to the Router Module environment.
 - The testbench connections to the Router Module UVC will need to be modified.
2. Test your changes in simulation.



Task 2: Using TLM Analysis FIFOs

The following guidance is intended to help you create a scoreboard implemented with TLM Analysis FIFOs:

1. Interface UVCs. The interface UVCs do not change. The UVC monitor analysis ports can be connected to either analysis FIFOs or analysis impconnectors. However, analysis FIFOs do not perform any cloning on input transactions. Therefore, you will need to check that the UVC monitors collect every transaction into a different instance to avoid overwriting data in the FIFOs.
2. Create a new scoreboard file with the following additions:
 - a. Instantiate analysis FIFOs for the YAPP, HBUS and all three Channel connections.
 - b. Instantiate getport connectors for each of the FIFO outputs and connect these to the get_peek_exportconnectors of the FIFO instantiations.
 - c. In a run-phase task, check the packets as follows:
 - Use a blocking getcall directly from the get_peek_exportconnector of the YAPP analysis FIFO instance to read input YAPP packets.
 - Discard the packet if the router is not enabled.
 - Discard the packet if it is not legal (length and address).
 - If the packet is valid, use a getcall to read from the get_peek_exportof the appropriate Channel analysis FIFO depending on the packet address.
 - Compare the packets and update any status counters.
 - In a concurrent forked method, read the HBUS analysis FIFO and on write operations, update local variables for maximum packet size and router enabled.
 - d. In a check phase method, update any status counters and check the FIFOs are empty.
 - e. In a report phase method, write out the status counte

