UVM Verification Project Report

SPI Core Wishbone Compatible

Ahmad Mukhtar

National University of Sciences and Technology (NUST)
Chip Design Centre (NCDC), Islamabad, Pakistan

November 28, 2024

Contents

1. TASK:1 Verification Plan

The following table outlines the comprehensive verification plan for the SPI Core design, covering all critical features and corresponding test cases:

Verification Plan Overview

This plan ensures a structured approach to validating SPI Core which Wishbone Compatible functionalities, with test cases designed for edge scenarios and regular operations.

Test Case ID	Test Name	Objective	Features to Verify	Stimulus	Expected Result
TC1	Reset Test	Verify control register reset values after asynchronous reset.	Directed (reset-only focus).	Upon reset, `empty` control registers is reset, and data pointers reset to initial states.	
TC2	Complete Read-/Write Test between Protocols	Verify correct read/write operations between Wishbone and SPI protocols.	Directed (cross-protocol verification).	DUT correctly reads and writes data between protocols. Wishbone master writes, SPI master reads and vice versa.	
TC3	Write Buffer Full Test	Verify system behavior when write buffer is full.	Directed (buffer overflow test).	System stops accepting writes when buffer is full, flags indicate full condition.	
TC4	Write Buffer Collision Test	Verify system's behavior during a buffer overflow/collision in write buffer.	Directed (overflow and collision handling).	System detects buffer overflow or collision and handles it correctly without data corruption.	
TC5	Read Buffer Full Test	Verify FIFO asserts `full` flag when buffer is full.	Directed (continuous writes).	`full` flag asserts when buffer is full, no further data can be written, write pointer stops incrementing.	

Table 1.1: Verification Plan for WB SPI

2. Usage and Verification Instructions

The following table provides clear, concise instructions on how to configure and execute the verification process for the WB SPI design:

Usage Instructions

Follow these steps to configure, run, and verify the FIFO design using the provided testbench.

Step	Action	Example

1. Set Parameters	Configure FIFO depth, data width, and test case number in `random_test`.	`depth = 16;`
		`width = 32;`
		`test = 4;`

2. Select Test Case	Choose the desired test case (1-7): **1**: Reset Test **2**: Synchronous Reset Test **3**: Depth Check Test **4**: Full Flag Test **5**: Empty Flag Test **6**: Write After Read Test **7**: Exhaustive Random Test	`test = 1` for Reset Test.
3. Run Simulation	Execute the testbench using your simulator (e.g., Cadence Xcelium).	`xrun -f file.f ...`
4. Check Results	Analyze logs and scoreboard outputs for Pass/Fail.	`Expected Data = 15,` `Actual Data = 15`
5. Verify Coverage	Review the coverage report to ensure 100% coverage of states and transitions in the test number 7.	Coverage metrics logged in the simulation tool.

Table 2.1: Usage and Verification Instructions for FIFO Design

2.1 DUT Register Model

2.1.1 DUT Register Model

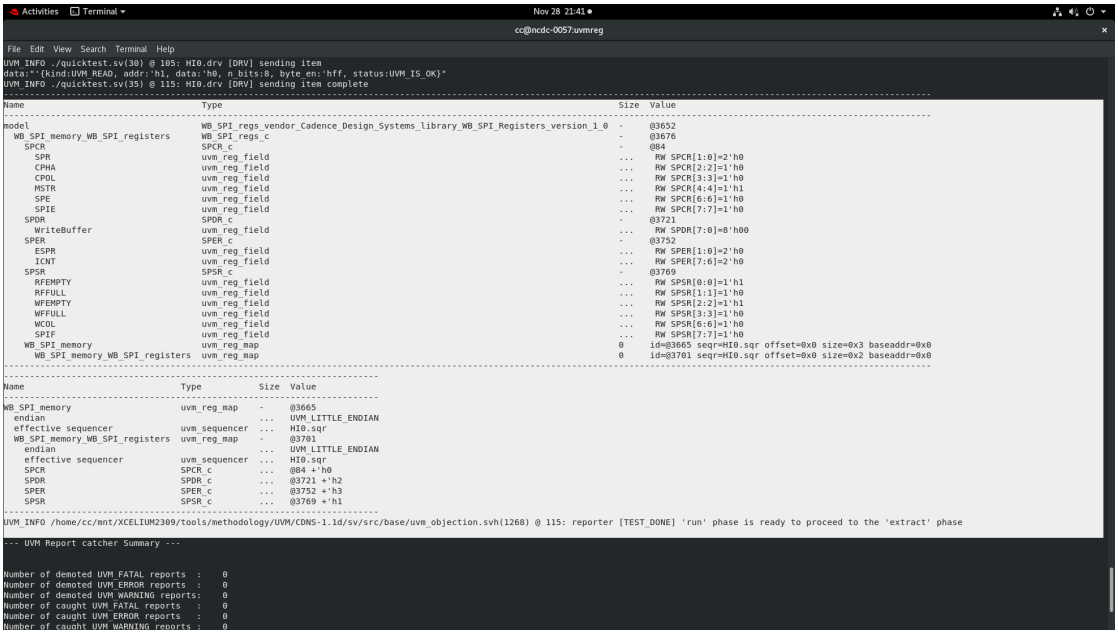

Figure 2.1: reseter model of the dut wb$_spi$.

3. Test Results

3.1 Test Results

3.1.1 Test Case 1: Reset Test

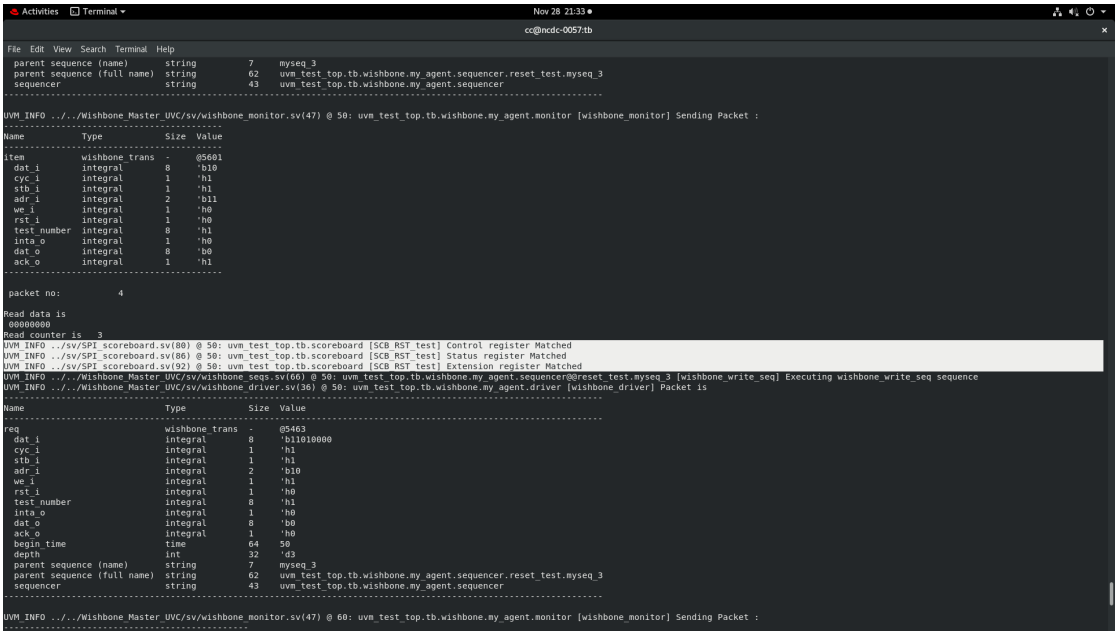

Figure 3.1: Simulation result for Reset Test.

3.1.2 Test Case 1: Reset Test error

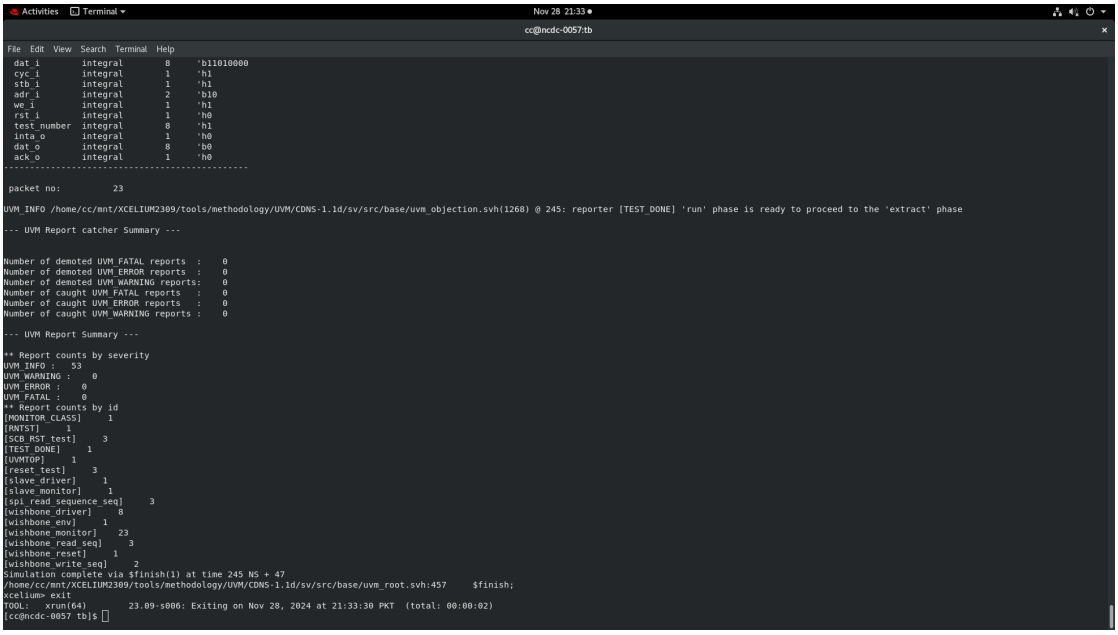

Figure 3.2: Simulation result for Reset Test.

3.1.3 Test Case 2: Complete read write between the protocols

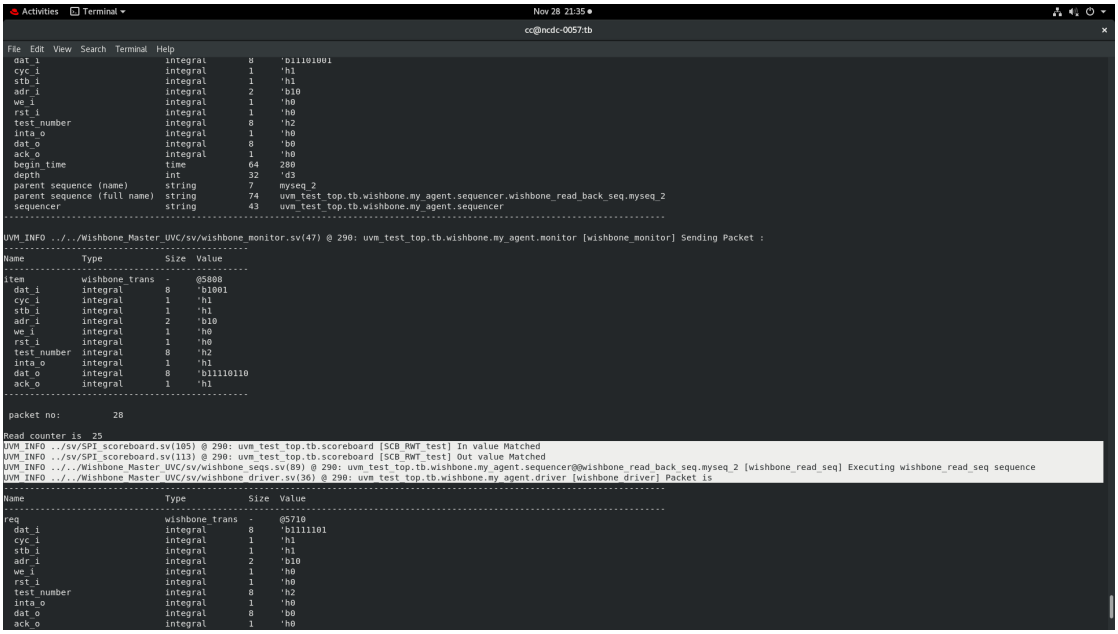

Figure 3.3: test_2.

3.1.4 Test Case 2: Complete read write between the protocols error

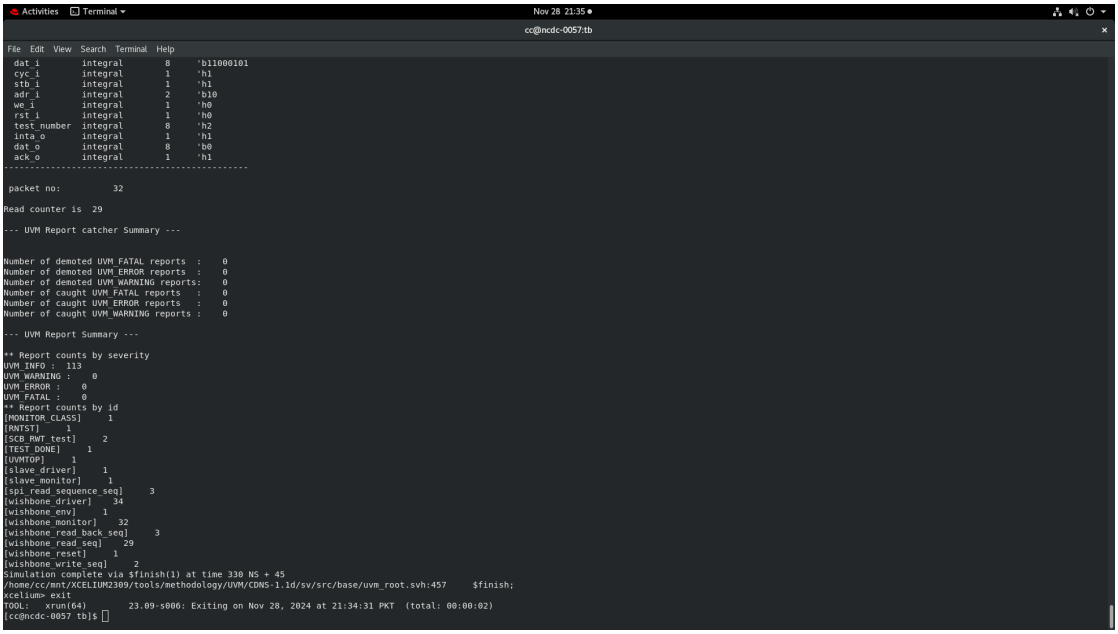

Figure 3.4: test_2.

3.1.5 $\mathbf{test}_2$

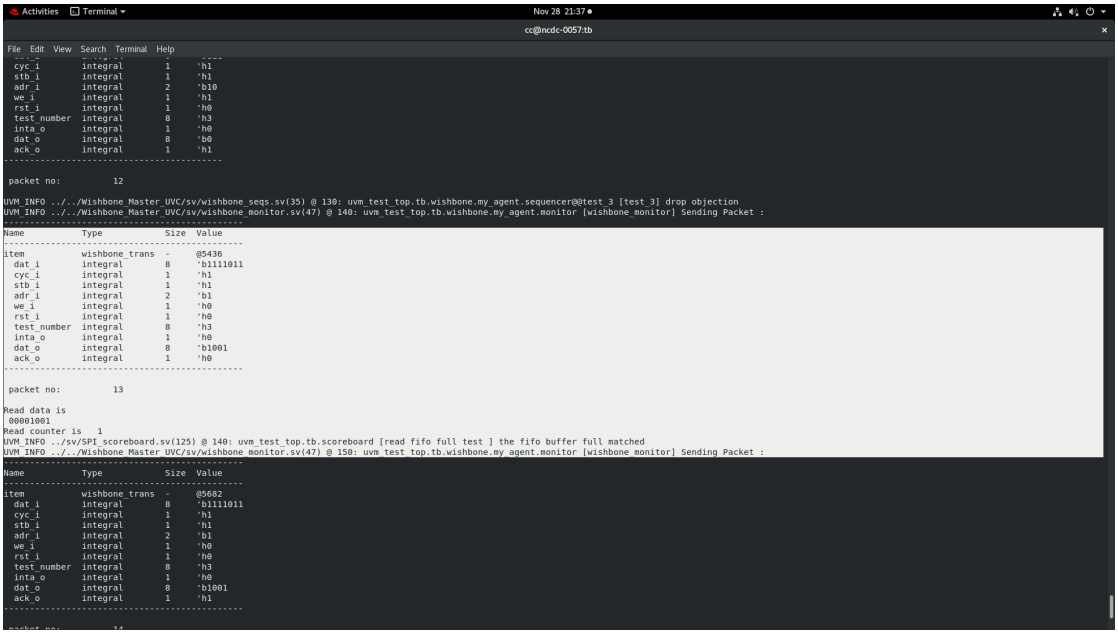

Figure 3.5: Simulation result for Depth Check Test.

3.1.6 test_3

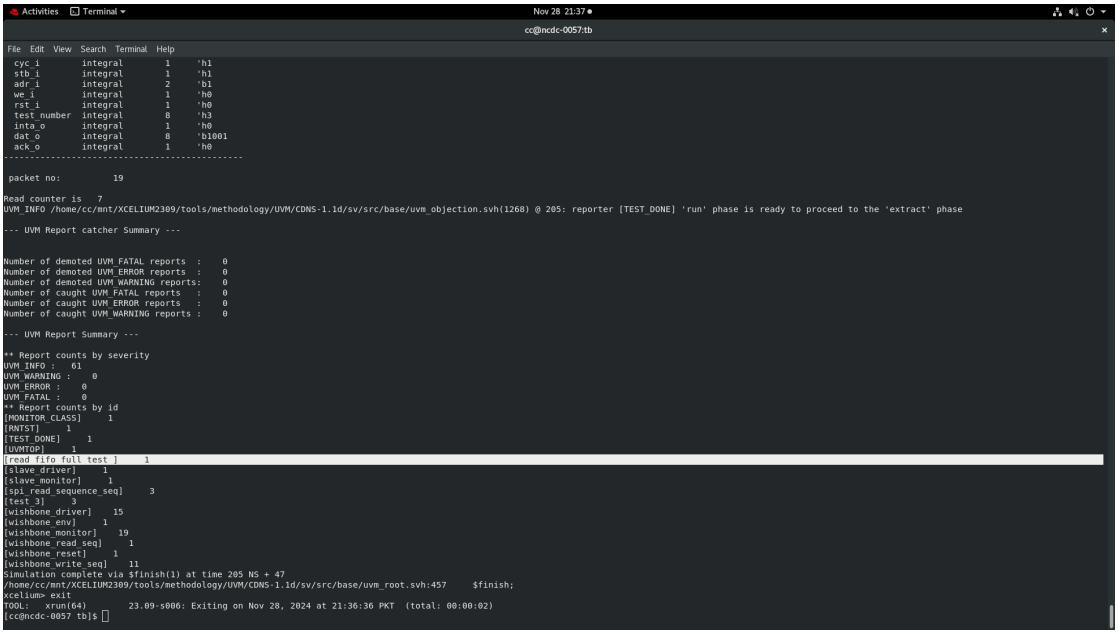

Figure 3.6: test_3.

3.1.7 Test Case 4: Write buffer Collision

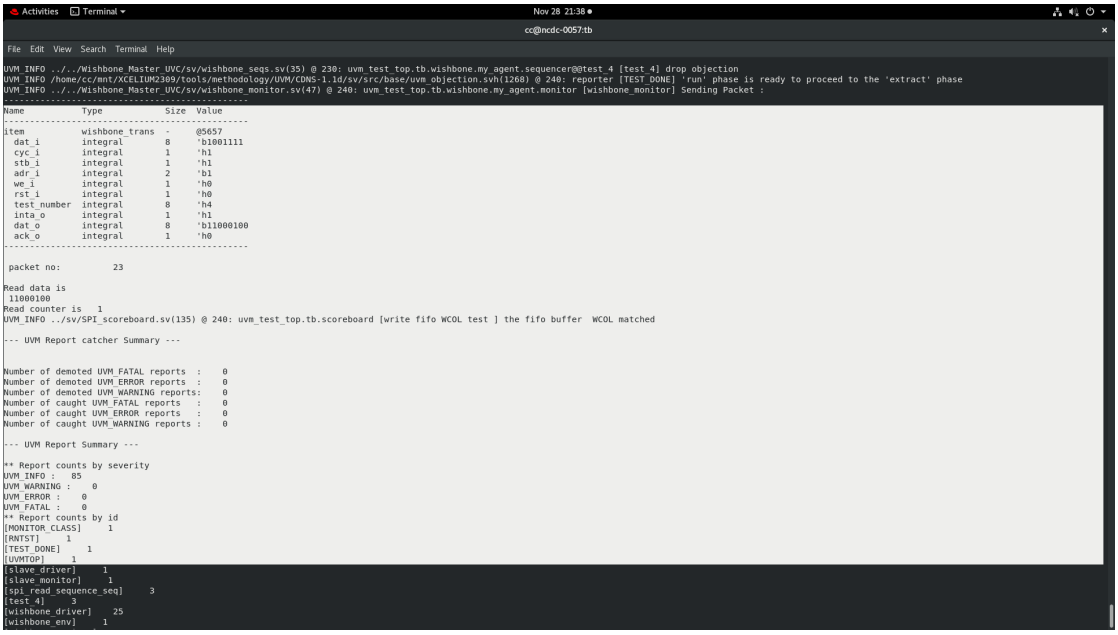

Figure 3.7: test_4.

3.1.8 Test Case 5: Read buffer Full

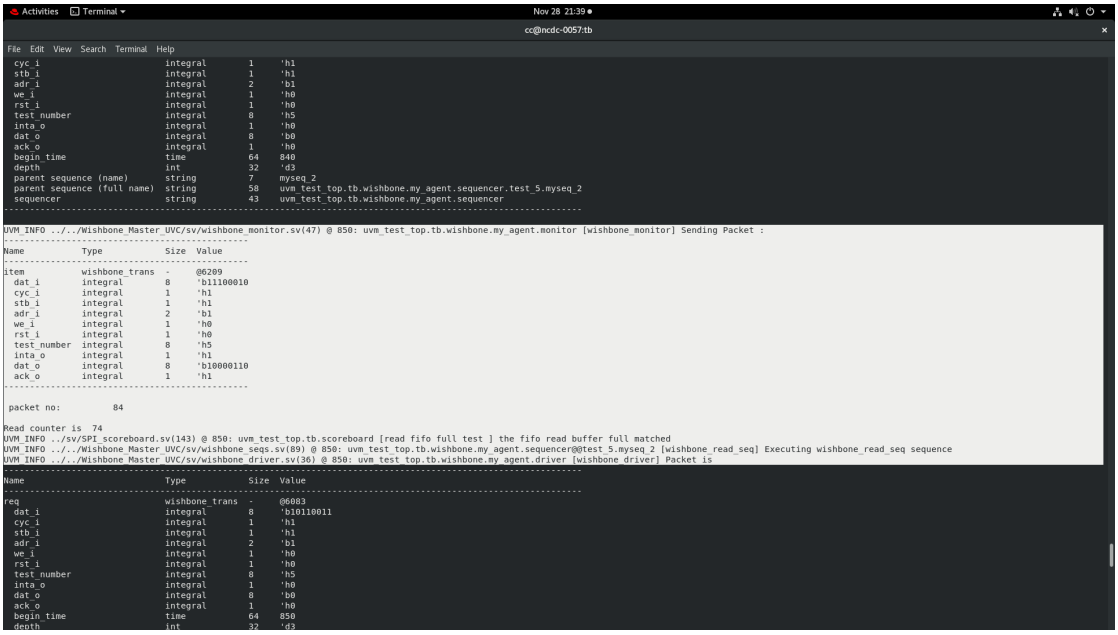

Figure 3.8: test 5.

3.1.9 Test Case 5: Read buffer Full

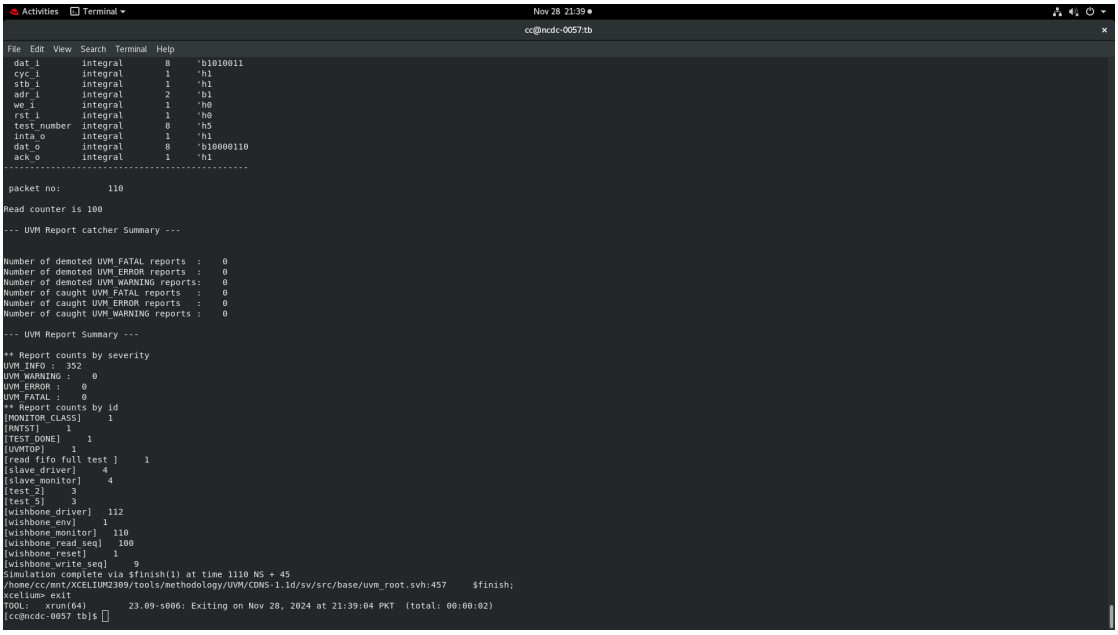

Figure 3.9: test_5.