Digital Design Verification

Lab Manual # 38 – Creating YAPP Interface UVM

Name: Ayesha Binte Safiullah
Submitted To: NCDC

Release: 1.0

Date: 6-Aug-2024

NUST Chip Design Centre (NCDC), Islamabad, Pakistan

Revision History

Revision Number	Revision Date	Revision By	Nature of Revision	Approved By
1.0	6/08/2024	Amber Khan	Complete Manual	-

Contents

Objective

The objectives of this lab are

- To the front end of a UVM Verification Component (UVC) and to explore the built-in phases of **`uvm_component`**.
- To create verification components and data using factory methods, and

to implement test classes using configurations.

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_lab10/
                ├── task1_uvc/
                │    ├── tb/
                │    │    ├── file.f
                │    │    ├── top.sv
                │    │    ├── router_tb.sv
                │    │    ├── router_test_lib.sv
                │    ├── sv/
                │    │    ├── yapp_pkj.sv
                │    │    ├── yapp_packet.sv
                │    │    ├── yapp_tx_env.sv
                │    │    ├── yapp_tx_agent.sv
                │    │    ├── yapp_tx_driver.sv
                │    │    ├── yapp_tx_monitor.sv
                │    │    ├── yapp_tx_sequencer.sv
                │    │    ├── yapp_tx_seqs.sv
                ├── task2_factory/
                │    ├── tb/
                │    │    ├── file.f
                │    │    ├── top.sv
                │    │    ├── router_tb.sv
                │    │    ├── router_test_lib.sv
                │    ├── sv/
                │    │    ├── yapp_pkj.sv
                │    │    ├── yapp_packet.sv
                │    │    ├── yapp_tx_env.sv
                │    │    ├── yapp_tx_agent.sv
                │    │    ├── yapp_tx_driver.sv
                │    │    ├── yapp_tx_monitor.sv
                │    │    ├── yapp_tx_sequencer.sv
                │    │    ├── yapp_tx_seqs.sv
```

Task 1: Creating a Simple UVC

You will be creating the driver, sequencer, monitor, agent and env for the UVC to drive the YAPP input port of the router. You will focus on the transmit (TX) agent for this task.

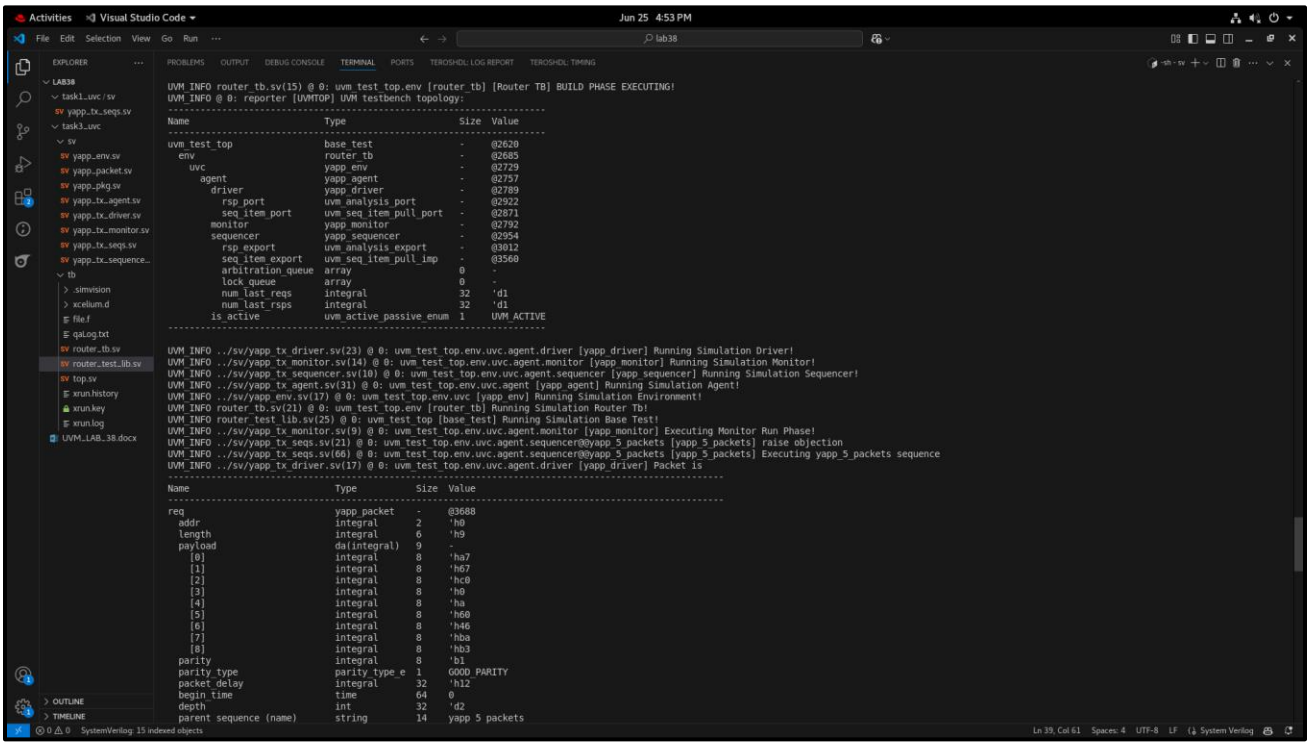
Checking the UVC Hierarchy

1. In the `task1_uvc/tb` directory, run a simulation using the `base_test` test class:
   a. Find the topology print.
   Does the hierarchy match your expectations?
   Answer: Yes.

   b. Use the topology print to find the full hierarchical pathname from your test class to your UVC sequencer (e.g., `tb.yapp.agent.sequencer`) and write it below.

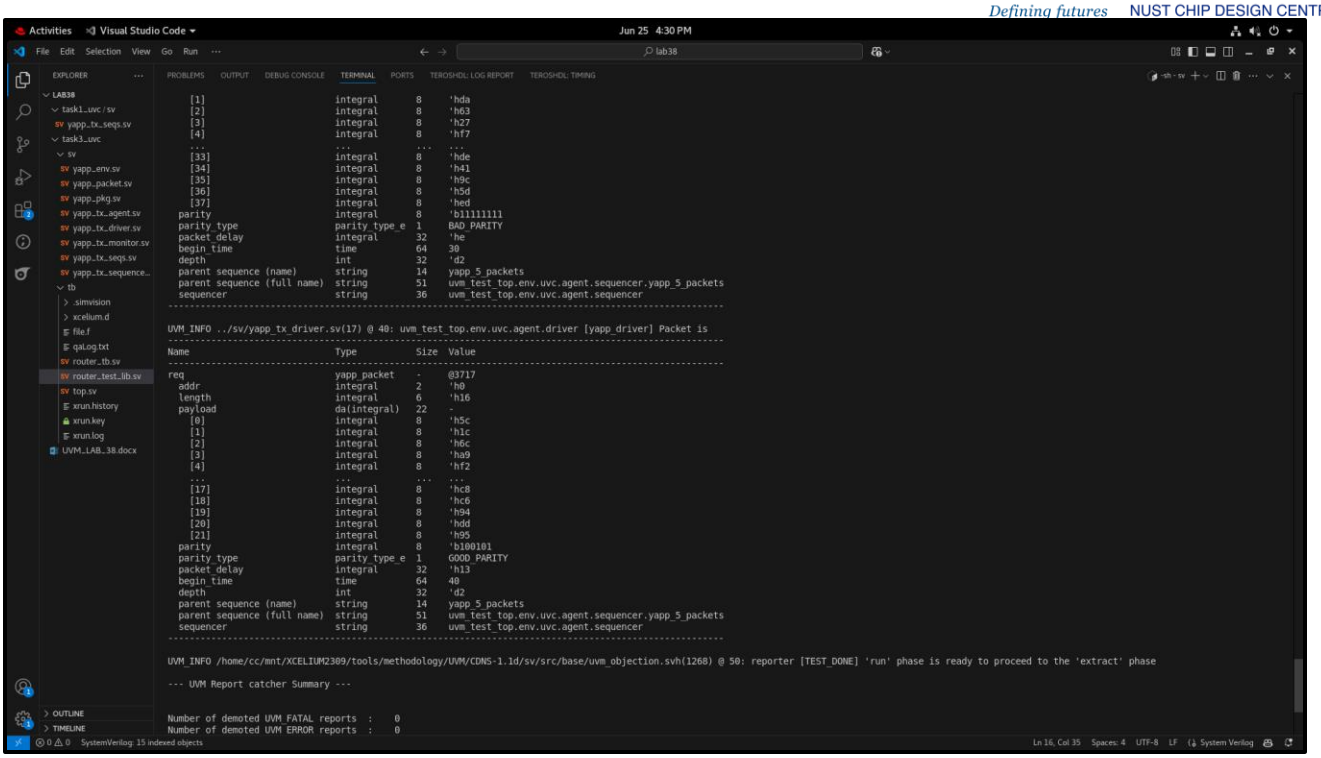
   Sequencer pathname: env.uvc.agent.sequencer

   c. Use your topology to find the value of the `is_active` property of the YAPP agent. *What is the value of the `is_active` variable when you printed the hierarchy?*
   Answer: UVM_ACTIVE



Running a Simple Sequence

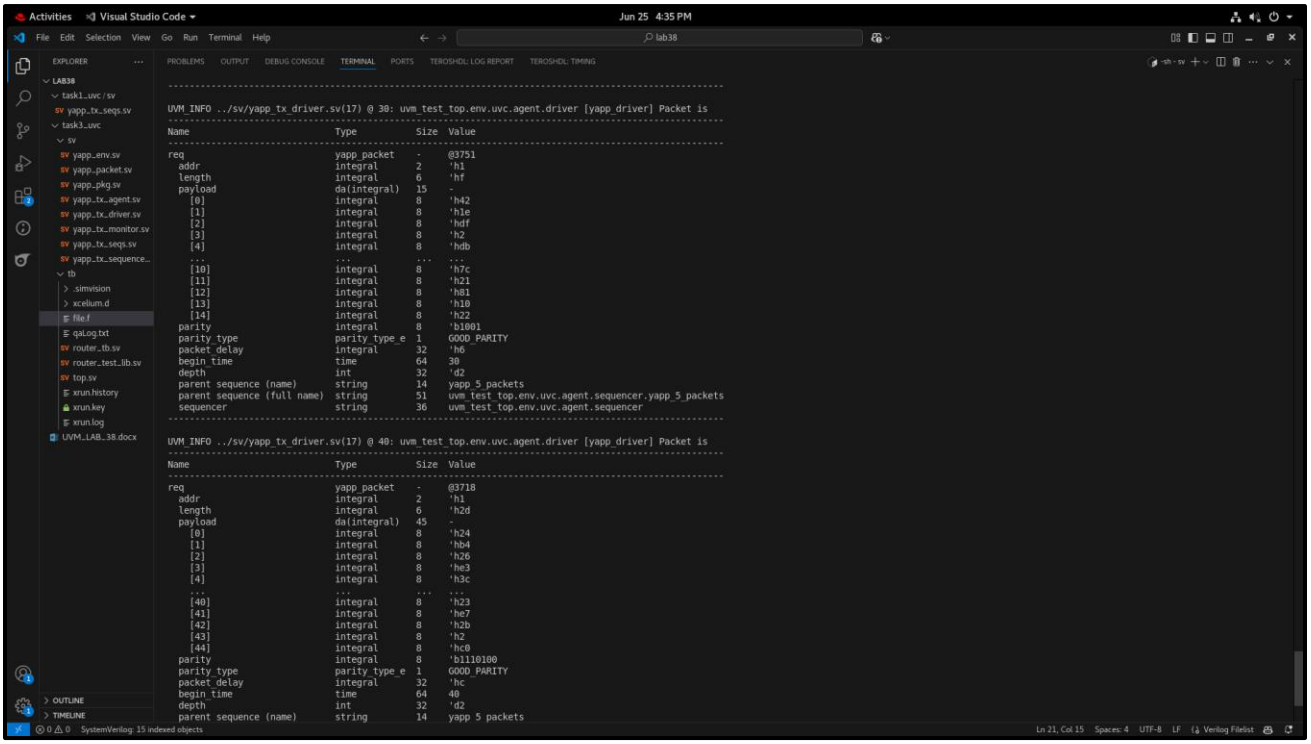
2. Run a simulation using the `base_test` test class:
   Your UVC should now generate and print YAPP packets. Check the correct number of packets are printed and every packet field is printed.



3. Add the following compilation option to the end of you command line:

   `+SVSEED=random`

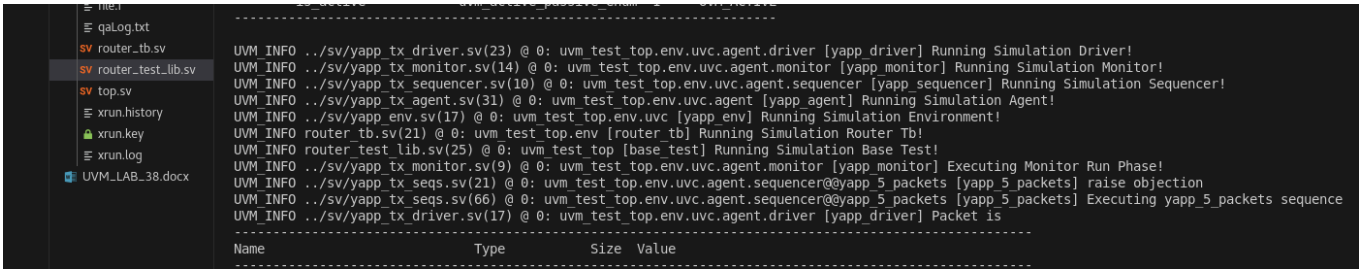
   This sets a random value for the initial randomization seed of the simulation. Re-run the Simulation(**do not recompile**) and you should see different packet data. The simulator reports the actual seed used for each simulation in the simulation log file.



4. Run a simulation with `base_test` and check which `start_of_simulation_phase()` method was called first. Which is called last?

Driver is called first; base test is called last.

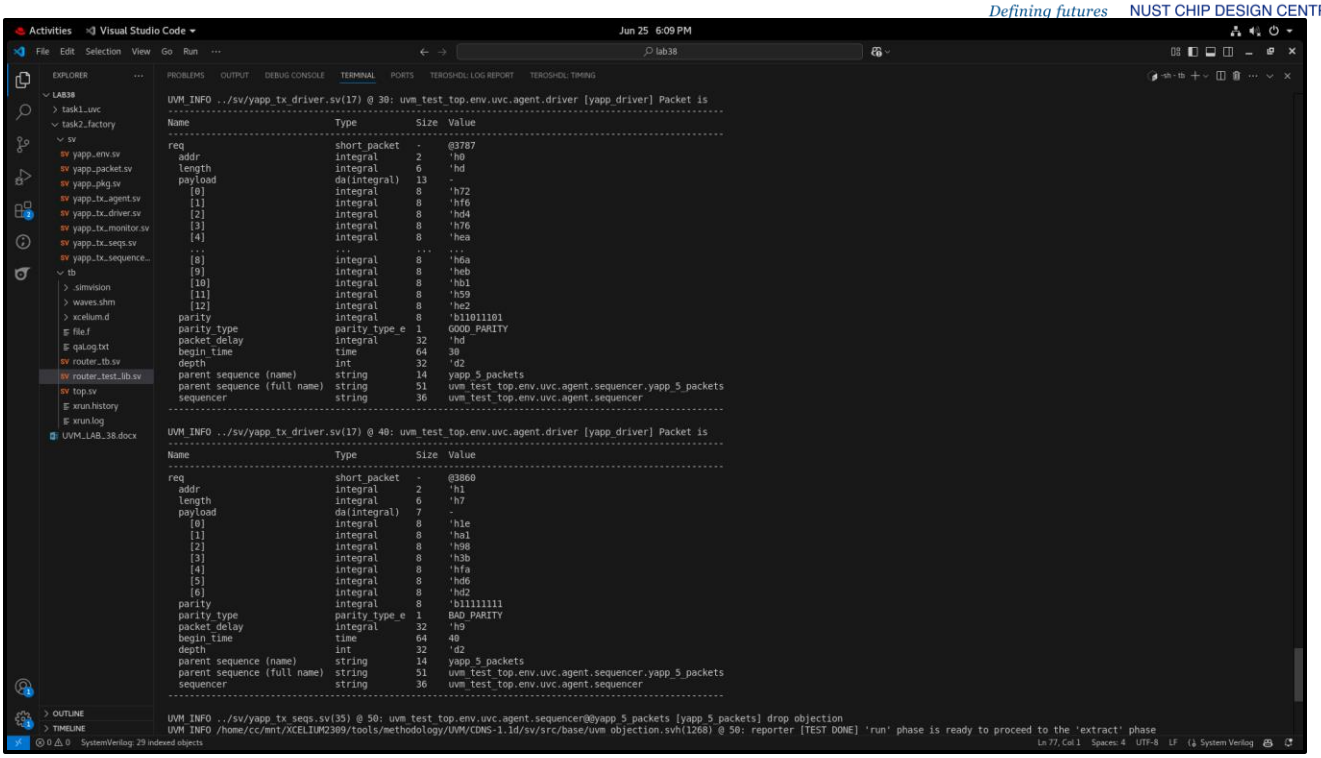
Why? You will need to set the right `+UVM_VERBOSITY` option to see the phase method messages.



Task 2: Using Factories

Using the Factory

1. Create a new short packet test as follows:
   a. Define a new packet type, short_yapp_packet, which extends from yapp_packet. Add this subclass definition to the end of your `sv/yapp_packet.sv` file.
   b. Add an object constructor and utility macro.
   c. Add a constraint in `short_yapp_packet` to limit packet length to less than 15.
   d. Add a constraint in `short_yapp_packet` to exclude an address value of 2.
   e. Define a new test, `short_packet_test`, in the file `router_test_lib.sv`. Extend this from `base_test`.
   f. In the `build_phase()` method of `short_packet_test`, use a `set_type_override` method to change the packet type to `short_yapp_packet`.
   g. Run the simulation using the new test, (+UVM_TESTNAME=`short_packet_test`), and check the correct packet type is created.



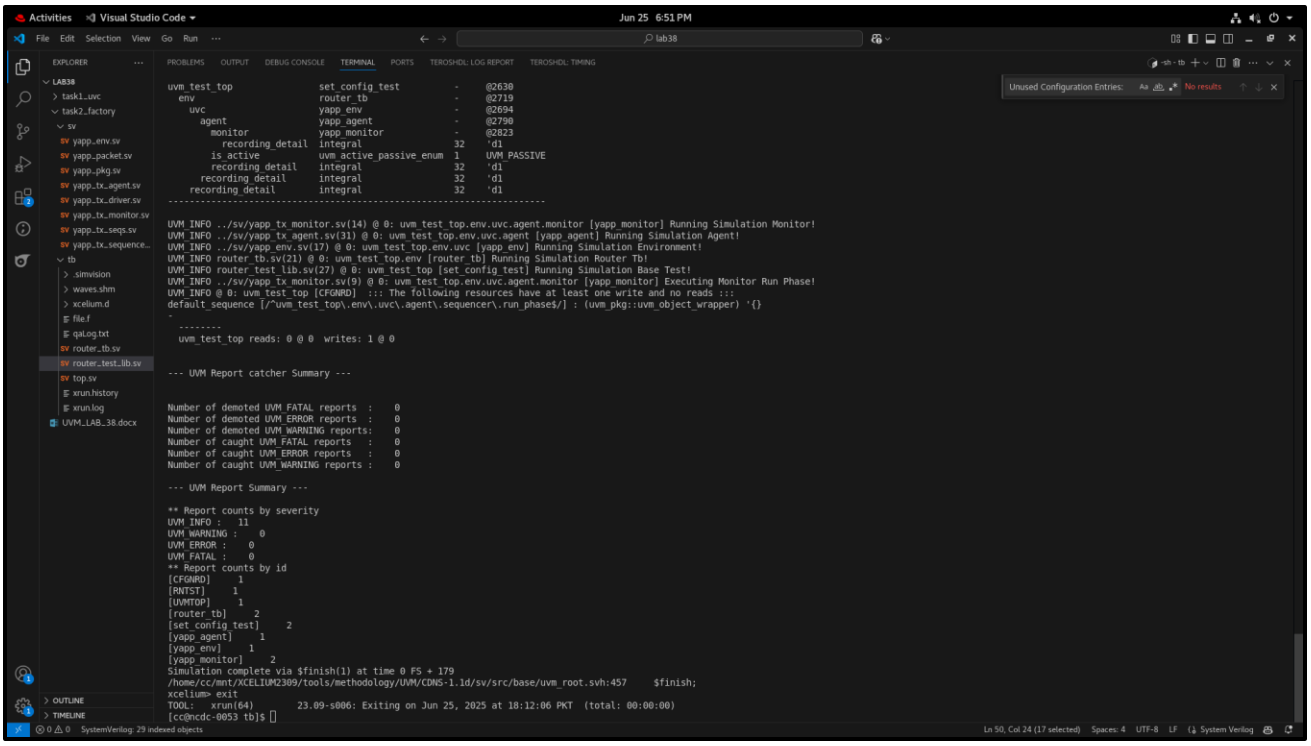
2. Create a new configuration test in the file `router_test_lib.sv`.
   a. Define a new test, `set_config_test`, which extends from `base_test`.
   b. In the `build_phase()` method, use a configuration method to set the `is_active` property of the YAPP TX agent to `UVM_PASSIVE`. Remember to call the configuration method before building the `yapp_env` instance.
   c. Run a simulation using the `set_config_test` test class (UVM_TESTNAME=`set_config_test`) and check the topology print to ensure your design is correctly configured.

   d. You should get a configuration usage report from `check_config_usage()`. Why do you get this?

   Answer: Because we have set the `is_active` property of the YAPP TX agent to `UVM_PASSIVE`.

   Although the configuration report maybe expected, it is good practice to minimize the number of reports where possible.

   Edit your test classes so that no configuration mismatch messages are reported, but all tests still work as required. Check your changes in simulation.


Activities
Visual Studio Code
Jun 25 6:51 PM
File Edit Selection View Go Run
lab38
EXPLORER
LAB38
task1_uvc
task2_factory
sv
yapp_env.sv
yapp_packet.sv
yapp_pkg.sv
yapp_tx_agent.sv
yapp_tx_driver.sv
yapp_tx_monitor.sv
yapp_tx_seqs.sv
yapp_tx_sequence...
tb
.simvision
waves.shm
xcelium.d
file.f
qaLog.txt
router_tb.sv
router_test_lib.sv
top.sv
xrun.history
xrun.log
UVM_LAB_38.docx
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TEROSHDL: LOG REPORT TEROSHDL: TIMING
Unused Configuration Entries:
No results
uvm_test_top set_config_test - @2630
env router_tb - @2719
uvc yapp_env - @2694
agent yapp_agent - @2790
monitor yapp_monitor - @2823
recording_detail integral 32 'd1
is_active uvm_active_passive_enum 1 UVM_PASSIVE
recording_detail integral 32 'd1
recording_detail integral 32 'd1
recording_detail integral 32 'd1
UVM_INFO ../sv/yapp_tx_monitor.sv(14) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Running Simulation Monitor!
UVM_INFO ../sv/yapp_tx_agent.sv(31) @ 0: uvm_test_top.env.uvc.agent [yapp_agent] Running Simulation Agent!
UVM_INFO ../sv/yapp_env.sv(17) @ 0: uvm_test_top.env.uvc [yapp_env] Running Simulation Environment!
UVM_INFO router_tb.sv(21) @ 0: uvm_test_top.env [router_tb] Running Simulation Router Tb!
UVM_INFO router_test_lib.sv(27) @ 0: uvm_test_top [set_config_test] Running Simulation Base Test!
UVM_INFO ../sv/yapp_tx_monitor.sv(9) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Executing Monitor Run Phase!
UVM_INFO @ 0: uvm_test_top [CFGNRD] ::: The following resources have at least one write and no reads :::
default_sequence [/^uvm_test_top\.env\.uvc\.agent\.sequencer\.run_phase$/] : (uvm_pkg::uvm_object_wrapper) '{}
uvm_test_top reads: 0 @ 0 writes: 1 @ 0
--- UVM Report catcher Summary ---
Number of demoted UVM_FATAL reports : 0
Number of demoted UVM_ERROR reports : 0
Number of demoted UVM_WARNING reports: 0
Number of caught UVM_FATAL reports : 0
Number of caught UVM_ERROR reports : 0
Number of caught UVM_WARNING reports : 0
--- UVM Report Summary ---
** Report counts by severity
UVM_INFO : 11
UVM_WARNING : 0
UVM_ERROR : 0
UVM_FATAL : 0
** Report counts by id
[CFGNRD] 1
[RNTST] 1
[UVMTOP] 1
[router_tb] 2
[set_config_test] 2
[yapp_agent] 1
[yapp_env] 1
[yapp_monitor] 2
Simulation complete via $finish(1) at time 0 FS + 179
/home/cc/mnt/XCELIUM2309/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_root.svh:457 $finish;
xcelium> exit
TOOL: xrun(64) 23.09-s006: Exiting on Jun 25, 2025 at 18:12:06 PKT (total: 00:00:00)
[cc@ncdc-0053 tb]$
SystemVerilog: 29 indexed objects
Ln 50, Col 24 (17 selected) Spaces: 4 UTF-8 LF System Verilog

THE END