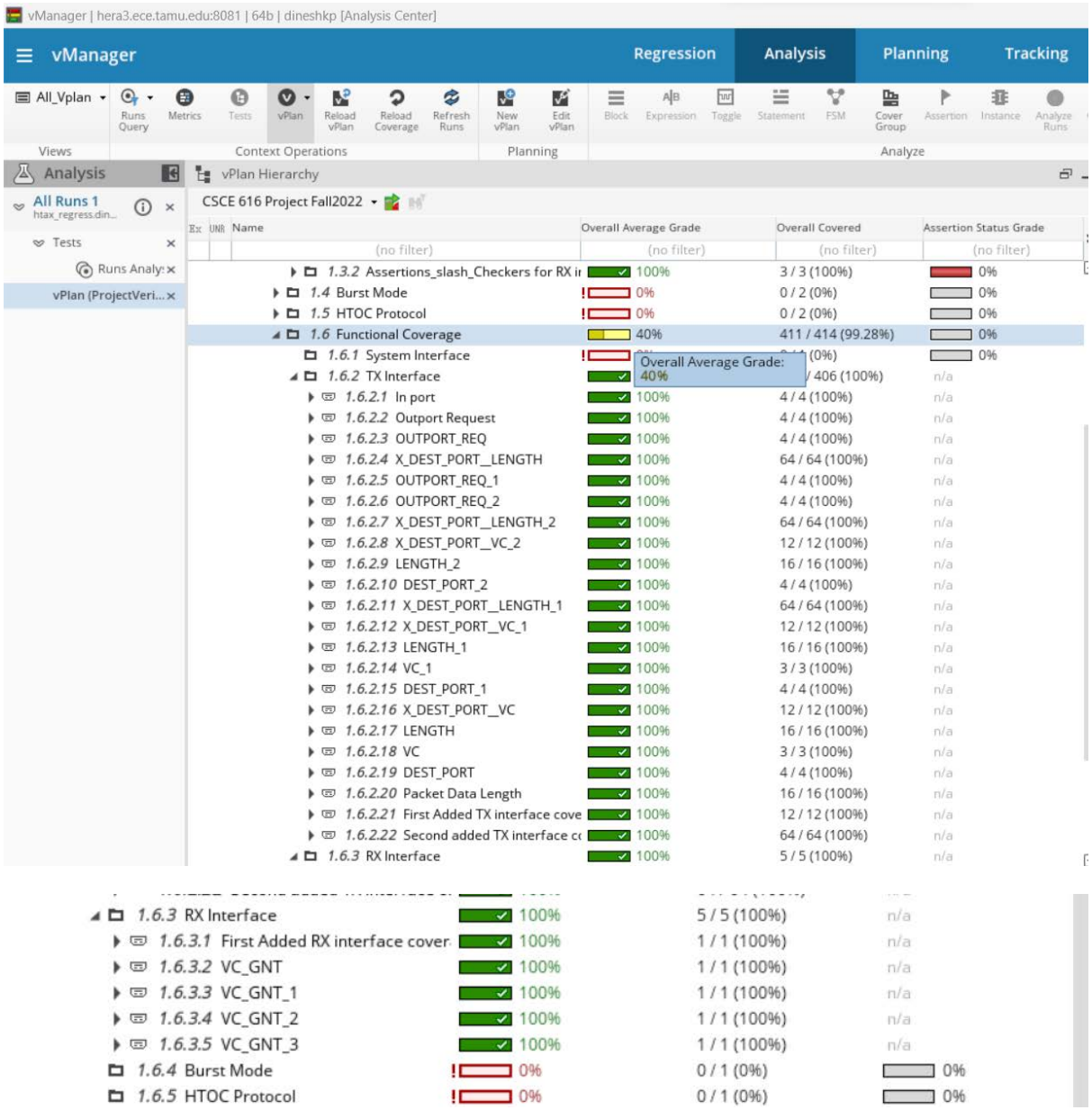
Lab 10 report

Coverage closure report:

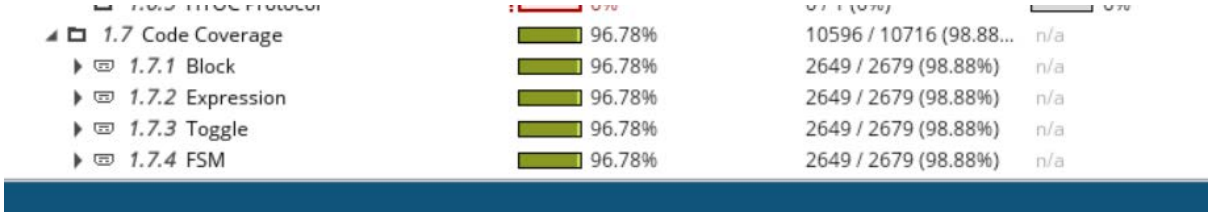
Functional coverage is 100% for both TX and RX interface

Initially 100% coverage was not achieved, in order to hit 100% coverage additional testcases were written. It can be seen that all possible values for the fields of packet like VC, tx_outport_req, length and dest_port are being covered.



Code coverage:

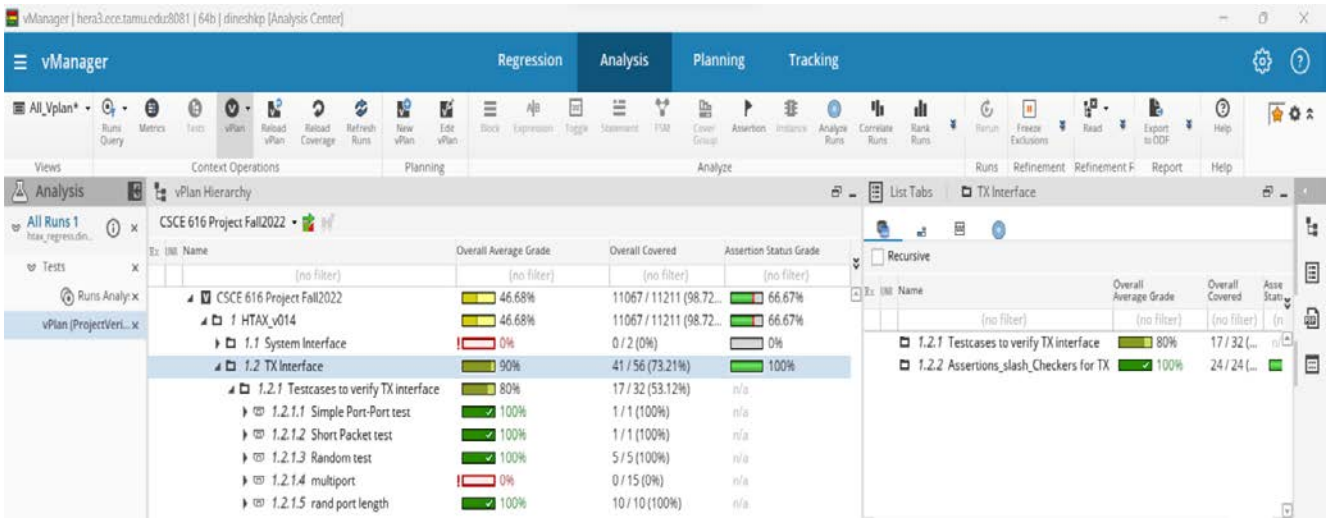
From the below screenshot, it can be seen that the code coverage is above 95% for block, toggle, expression, and FSM.

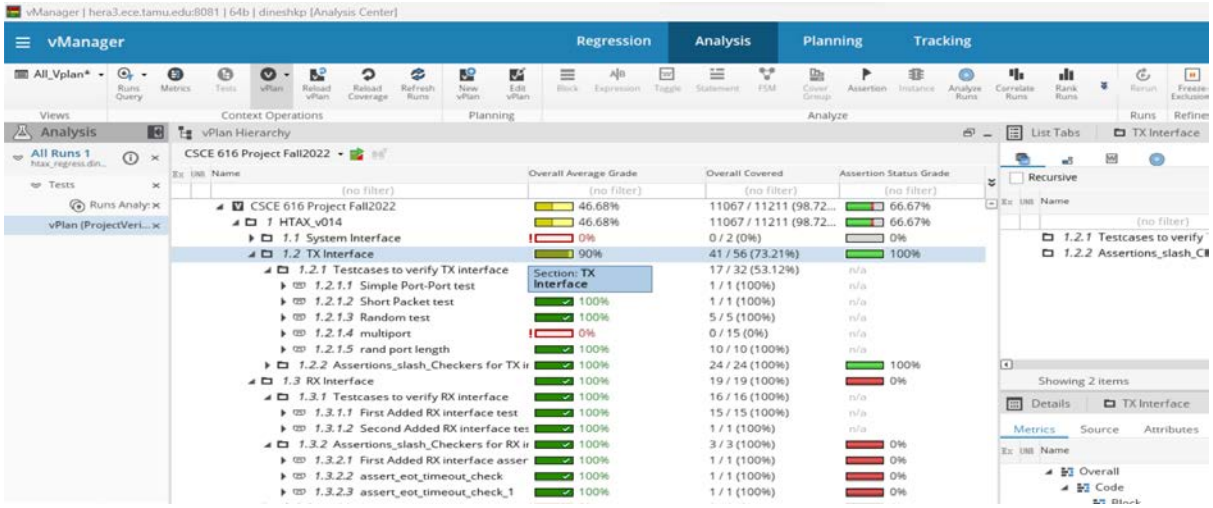


Bug report:

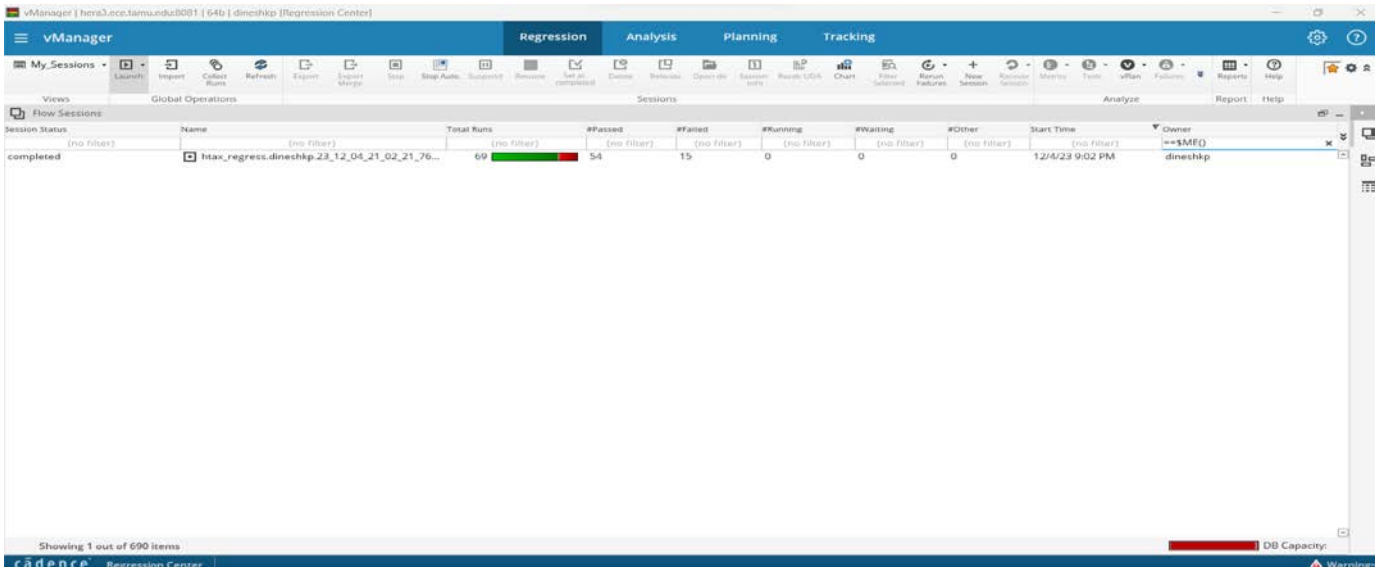
Additional tests were included to identify the bug. The htax_random_test file contains the testcases that were used to identify the bug in the design. In this, we generate packets to all the 4 ports parallel using fork-join. Additional constraints were written such that all dest_ports are also covered. Also, packet length and delay has to be kept constant for all 4 packets to hit this bug.

Testcases for TX and RX interface in Vmanager:





In the above figure, only the testcase that contains the bug tests fails.



The test that includes the testcases for finding the bug has been called 15 times. So these tests 15 fails and other tests pass.

The above test is executed using the irun command and the assertion fails.

```
xmsim: *F,ASRTST (../tb/htax_rx_interface.sv,56): (time 20430 NS) Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed
Memory Usage - Current physical: 91.2M, Current virtual: 175.1M
CPU Usage - 0.2s system + 0.2s user = 0.4s total (62.7% cpu)
Simulation terminated via $fatal(2) at time 20430 NS + 2
../tb/htax_rx_interface.sv:56       $fatal("HTAX_RX_INF ERROR : TIMEOUT rx_eot did not occur within 1000 cycles after rx_sot");
xcelium> exit

coverage setup:
  workdir  :  ./cov_work
  dutinst  :  top(top)
  scope    :  scope
  testname :  test

coverage files:
  model(design data) :  ./cov_work/scope/icc_4e8e3c4e_7afbaf4d.ucm (reused)
  data               :  ./cov_work/scope/test/icc_4e8e3c4e_7afbaf4d.ucd
TOOL:   xrun    22.03-s004: Exiting on Dec 04, 2023 at 21:21:09 CST  (total: 00:00:07)
```

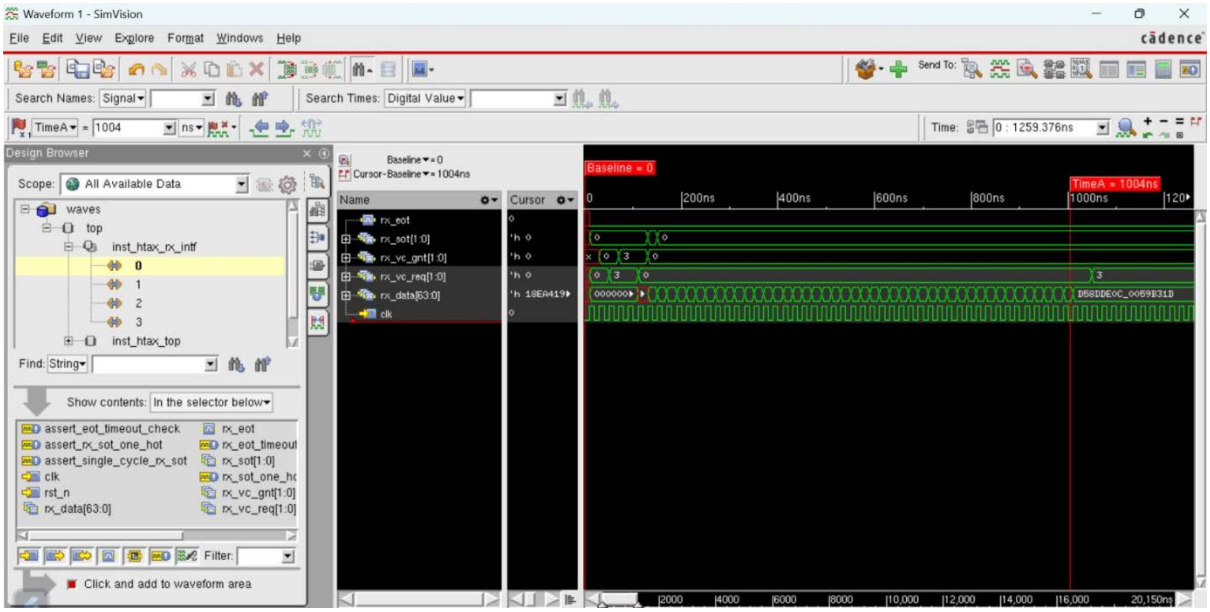
Bug in the code:

When packets are generated in all ports the eot_in become 1. The complement of it produces a 0 and thus the AND with it makes eot always 0.

```
always @( * )
begin
        (* full_case *) (* parallel_case *)
        casex (inport_sel)
                4'b1xxx: selected_sot = sot_in[((4*VC)-1):(3*VC)];
                4'bx1xx: selected_sot = sot_in[((3*VC)-1):(2*VC)];
                4'bxx1x: selected_sot = sot_in[((2*VC)-1):(1*VC)];
                4'bxxx1: selected_sot = sot_in[((1*VC)-1):(0*VC)];
        endcase
end

assign selected_eot = |(eot_in & inport_sel_reg) & ~(&(eot_in));
```

Waveform with eot not asserted:



From the above waveform, it is clear that the rx_eot signal is not asserted for any packet that is transmitted. But, it is supposed to be asserted when the final packet is transmitted for each transaction. Tracing this bug in the design, it was found that a NOT of bitwise AND was performed on eot_in.

Due to this, whenever all the ports are accessed, all the bits become high and the bitwise AND becomes 1. Hence, a complement of this will give a 0. So the EOT signal is 0 at all times.

To fix this, we can remove the bitwise AND part and this fixes the bug.

Fixed code:

```
always @( * )
begin
        (* full_case *) (* parallel_case *)
        casex (inport_sel)
                4'b1xxx: selected_sot = sot_in[((4*VC)-1):(3*VC)];
                4'bx1xx: selected_sot = sot_in[((3*VC)-1):(2*VC)];
                4'bxx1x: selected_sot = sot_in[((2*VC)-1):(1*VC)];
                4'bxxx1: selected_sot = sot_in[((1*VC)-1):(0*VC)];
        endcase
end

assign selected_eot = |(eot_in & inport_sel_reg);
```

After fixing the code the UVM fatal does not occur:

```
--- 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 :  136
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
```

EOT gets asserted in the waveform after fixing the bug: