**Coverage report**

**Failing Regression:**

A screenshot of a computer

Description automatically generated

**Passing Regression after the failing regression:**

A screenshot of a computer

Description automatically generated

**TestCases Mapped:**

A screenshot of a computer

Description automatically generated

**Assertions/Checkers for Tx interface:**

A screenshot of a computer

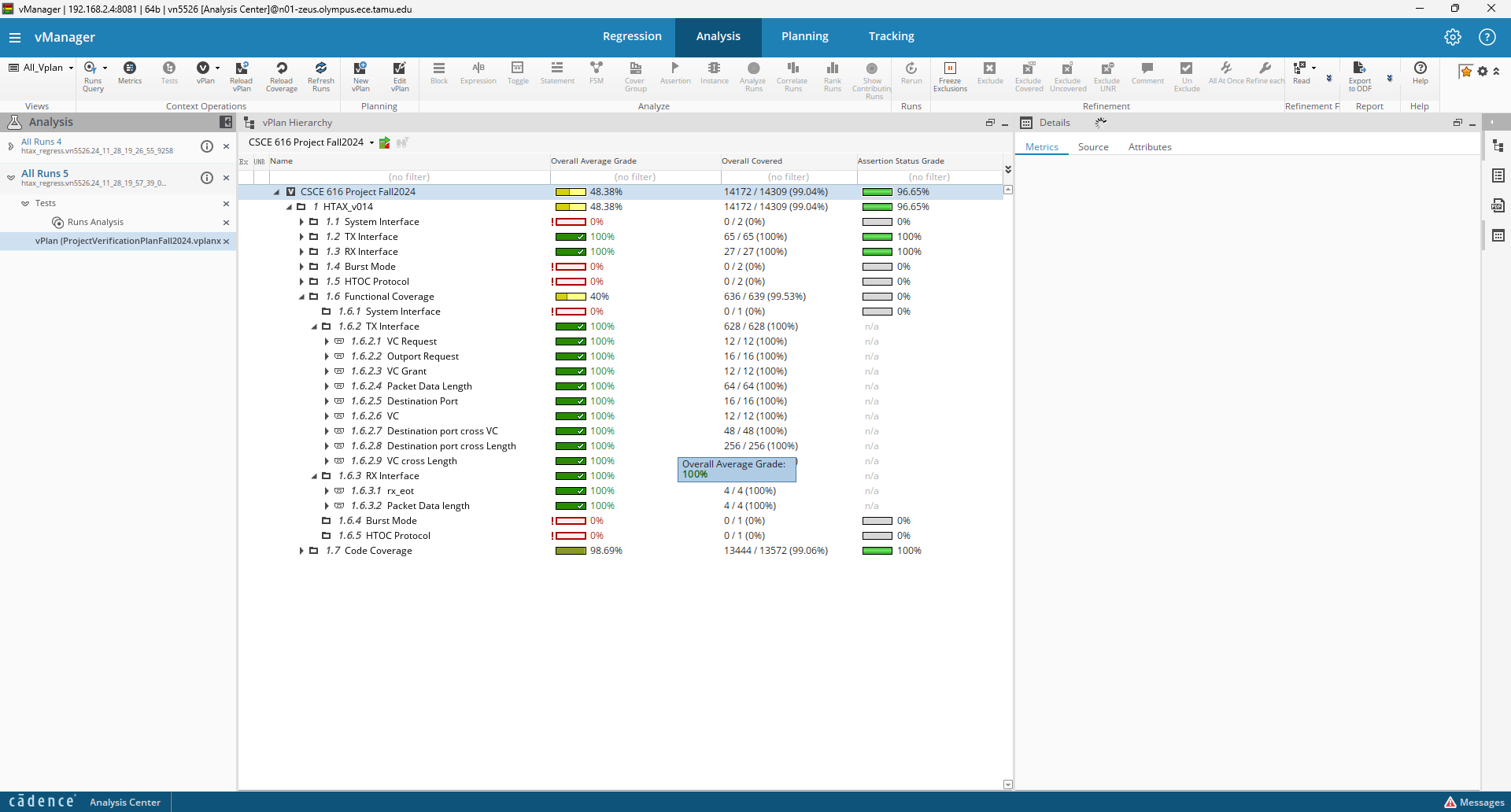
Description automatically generated

**Assertions/Checkers for Rx interface:**

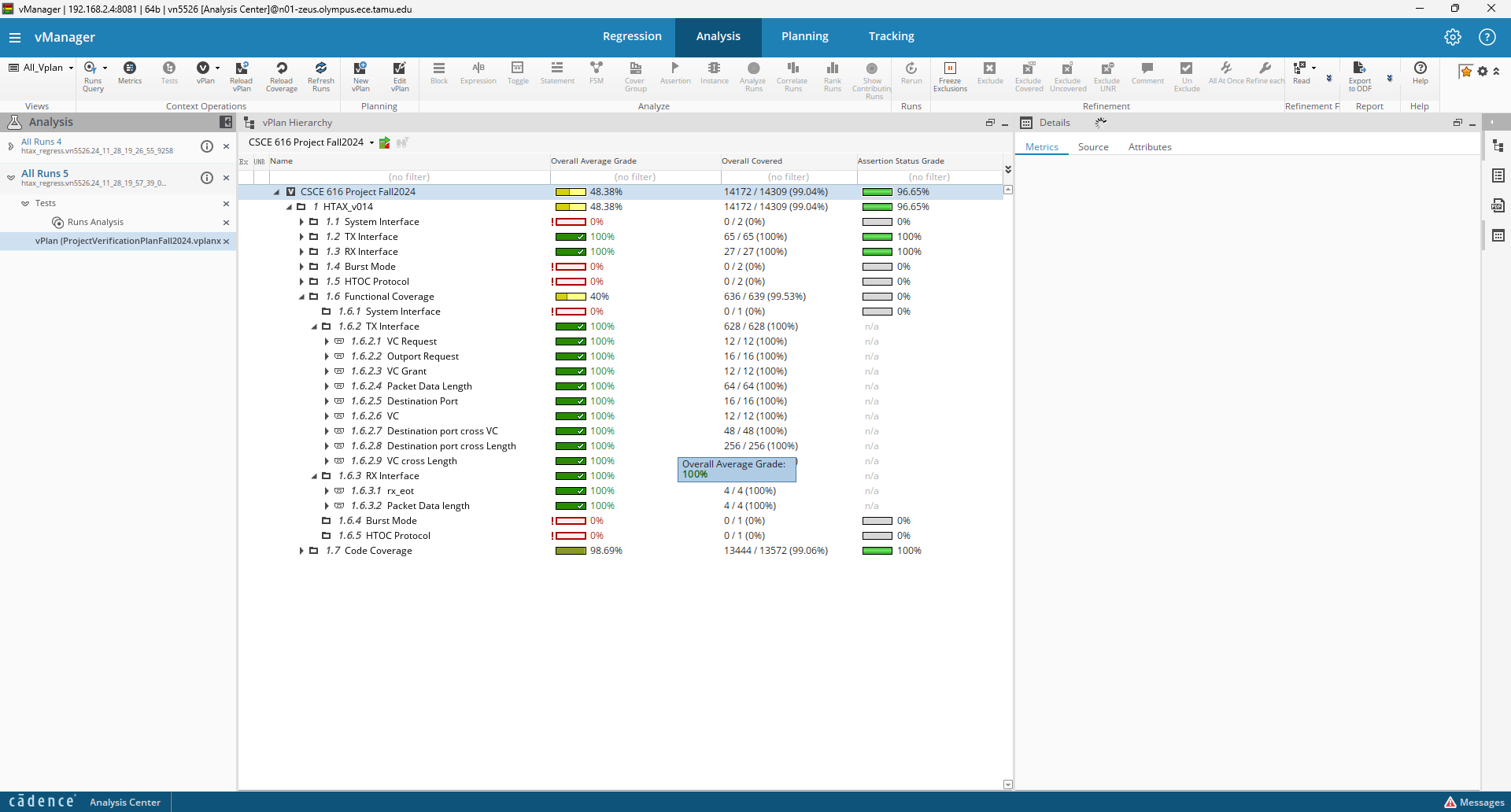
A screenshot of a computer

Description automatically generated

**Functional Coverage for Tx interface:**



**Functional Coverage for Rx interface:**



**Code Coverage:**

A screenshot of a computer

Description automatically generated

**Code Coverage Holes:**

[Explain any code coverage holes (if any)]

**Bugs report**

**Bug 1**

**What is the bug?**

When multiple short packet sequences are received parallelly on different ports, the rx\_eot\_timeout\_check assertion fails. This means that End of Transmission signal is creating an issue in some part of the design files.

**Where is it?**

Module: module htax\_outport\_data\_mux

File: module htax\_outport\_data\_mux.v

Line number(s): L43

**How to reproduce:**



By running this parallelized short-packet test, the assertion bug can be triggered.

**Expected behavior:**

The selected\_eot signal in the DUT should correctly indicate the End of Transmission. This should work even if EOT occurs on one or more than one input port. This in turn drives the eot timeout assertion which has a tran\_cycles\_left counter which should decrement when tran\_on is active.

**Actual behavior:**

assign selected\_eot = |(eot\_in & inport\_sel\_reg) & ~(&(eot\_in));

The selected\_eot signal does not assert when all eot\_in bits are set. As a result, the tran\_cycles\_left counter does not decrement and leads to the eot\_timeout assertion fails.

**Bug fix:**

Removing the & ~(&(eot\_in)) from the design file, should fix the code.

assign selected\_eot = |(eot\_in & inport\_sel\_reg);

This way, the selected\_eot signal asserts if any valid EOT signal exists. This removes the erroneous dependency on ~(&(eot\_in)), which incorrectly deasserted selected\_eot when all eot\_in bits were set.

**Failing Assertion:**

A screenshot of a computer

Description automatically generated

**Failing Scenario Waveform:**

A screenshot of a computer

Description automatically generated

**Failing Assertion Passing after the fix:**

A screenshot of a computer

Description automatically generated