Lab report

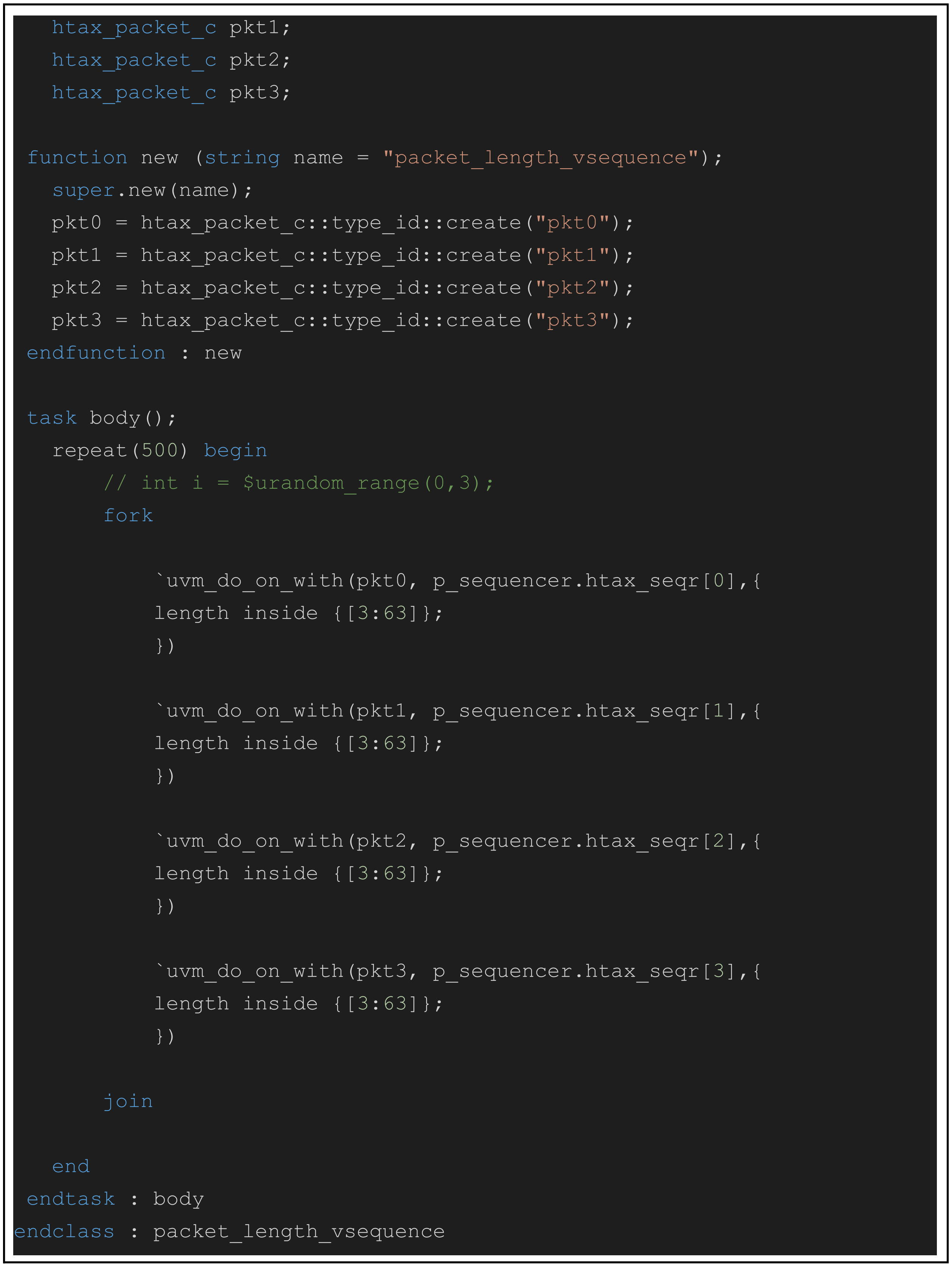
Code

```
lass parallel_random_vsequence extends htax_base_vseq;

 `uvm_object_utils(parallel_random_vsequence)

 htax_packet_c pkt0;
 htax_packet_c pkt1;
 htax_packet_c pkt2;
 htax_packet_c pkt3;

 function new (string name = "parallel_random_vsequence");
   super.new(name);
   pkt0 = htax_packet_c::type_id::create("pkt0");
   pkt1 = htax_packet_c::type_id::create("pkt1");
   pkt2 = htax_packet_c::type_id::create("pkt2");
   pkt3 = htax_packet_c::type_id::create("pkt3");
 endfunction : new

 // Helper function to generate a random permutation
 function void get_random_permutation(ref int ports[$]);
   int temp, rand_idx;
   for (int i = ports.size()-1; i > 0; i--) begin
     rand_idx = $urandom_range(0, i);
     temp = ports[i];
     ports[i] = ports[rand_idx];
     ports[rand_idx] = temp;
   end
 endfunction : get_random_permutation

 task body();
   int ports[$] = '{0, 1, 2, 3}; // Array of port indices

   repeat(200) begin
     get_random_permutation(ports); // Generate a random permutation of
ports
```

```
     fork
       `uvm_do_on_with(pkt0, p_sequencer.htax_seqr[0], {
         pkt0.dest_port == ports[0];
         pkt0.length inside {[3:10]};
         pkt0.delay < 10;
       })

       `uvm_do_on_with(pkt1, p_sequencer.htax_seqr[1], {
         pkt1.dest_port == ports[1];
         pkt1.length inside {[3:10]};
         pkt1.delay < 10;
       })

       `uvm_do_on_with(pkt2, p_sequencer.htax_seqr[2], {
         pkt2.dest_port == ports[2];
         pkt2.length inside {[3:10]};
         pkt2.delay < 10;
       })

       `uvm_do_on_with(pkt3, p_sequencer.htax_seqr[3], {
         pkt3.dest_port == ports[3];
         pkt3.length inside {[3:10]};
         pkt3.delay < 10;
       })
     join
   end
 endtask : body

endclass : parallel_random_vsequence
```

Parallel random test (the one which triggered the bug)

```
class packet_length_vsequence extends htax_base_vseq;

 `uvm_object_utils(packet_length_vsequence)

   htax_packet_c pkt0;
```



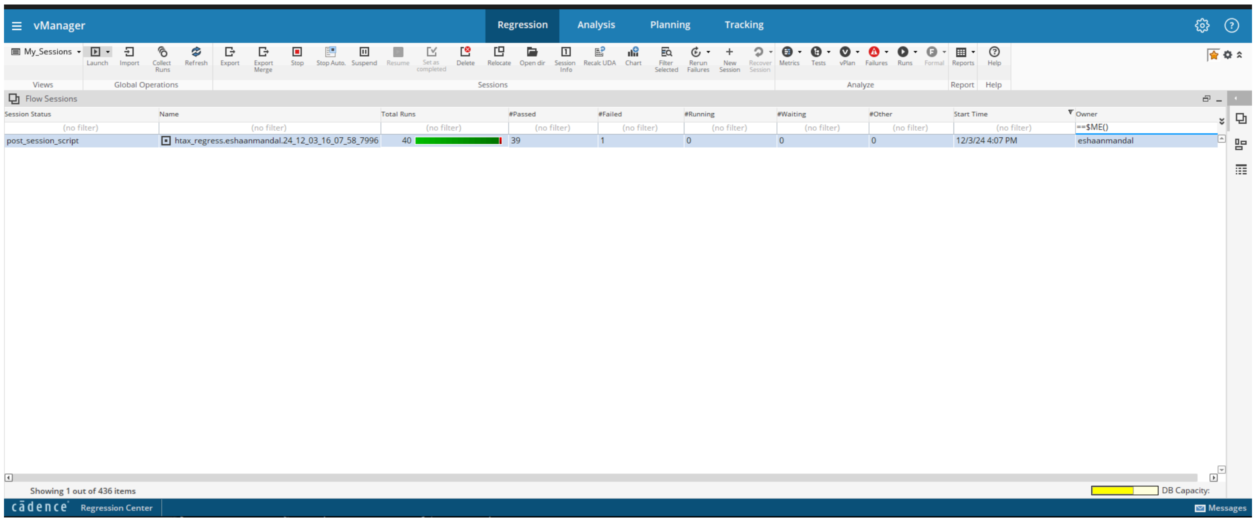
```
   htax_packet_c pkt1;
   htax_packet_c pkt2;
   htax_packet_c pkt3;

 function new (string name = "packet_length_vsequence");
   super.new(name);
   pkt0 = htax_packet_c::type_id::create("pkt0");
   pkt1 = htax_packet_c::type_id::create("pkt1");
   pkt2 = htax_packet_c::type_id::create("pkt2");
   pkt3 = htax_packet_c::type_id::create("pkt3");
 endfunction : new

 task body();
   repeat(500) begin
       // int i = $urandom_range(0,3);
       fork

           `uvm_do_on_with(pkt0, p_sequencer.htax_seqr[0],{
           length inside {[3:63]};
           })

           `uvm_do_on_with(pkt1, p_sequencer.htax_seqr[1],{
           length inside {[3:63]};
           })

           `uvm_do_on_with(pkt2, p_sequencer.htax_seqr[2],{
           length inside {[3:63]};
           })

           `uvm_do_on_with(pkt3, p_sequencer.htax_seqr[3],{
           length inside {[3:63]};
           })

       join

   end
 endtask : body
endclass : packet_length_vsequence
```

Variable length test

```
class variable_delay_vsequence extends htax_base_vseq;

 `uvm_object_utils(variable_delay_vsequence)

//   htax_packet_c pkt0;
//   htax_packet_c pkt1;
//   htax_packet_c pkt2;
//   htax_packet_c pkt3;

 function new (string name = "variable_delay_vsequence");
   super.new(name);
 endfunction : new

 task body();
   repeat(500) begin
       int i = $urandom_range(0,3);
       `uvm_do_on_with(req, p_sequencer.htax_seqr[i],{
           delay inside {[1:20]};
       })
   end
 endtask : body
endclass : variable_delay_vsequence
```

Variable delay test

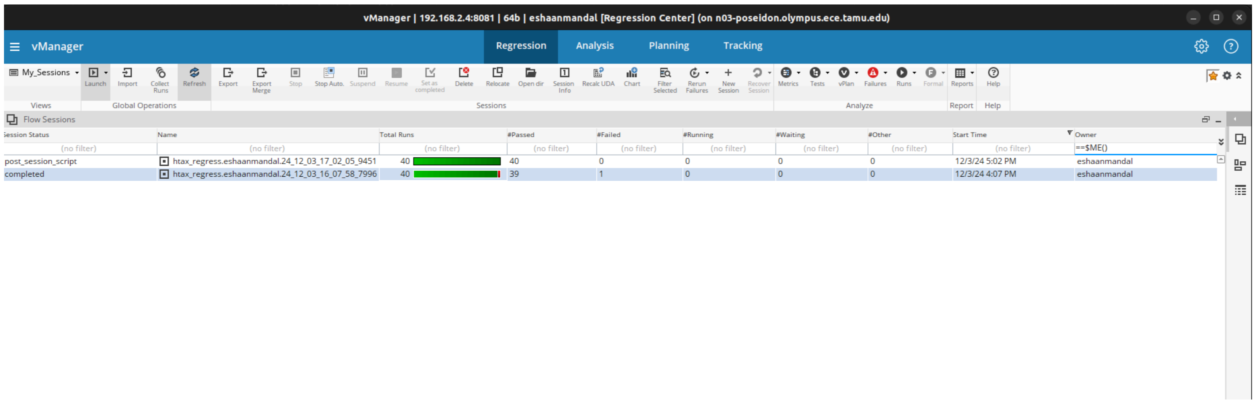
Failing Regression:



Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	==$ME()
post_session_script	htax_regress.eshaanmandal.24_12_03_16_07_58_7996	40	39	1	0	0	0	12/3/24 4:07 PM	eshaanmandal

Showing 1 out of 436 items

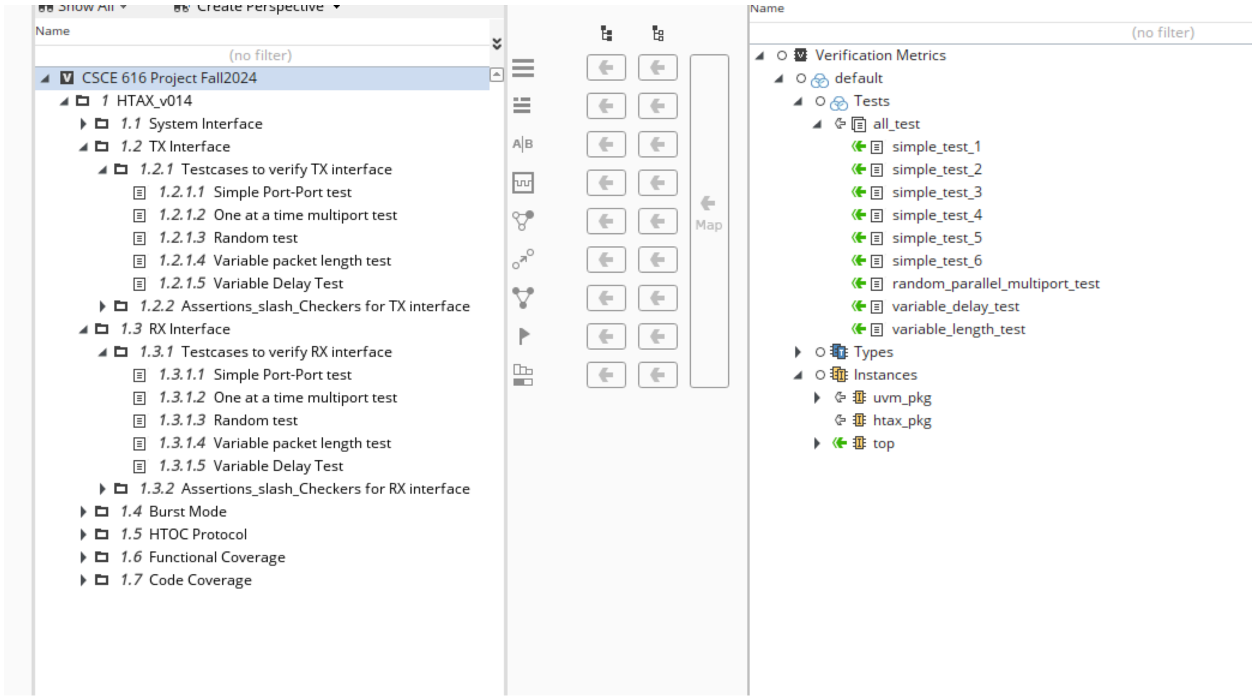
Passing Regression after the failing regression:



vManager | 192.168.2.4:8081 | 64b | eshaanmandal [Regression Center] (on n03-poseidon.olympus.ece.tamu.edu)

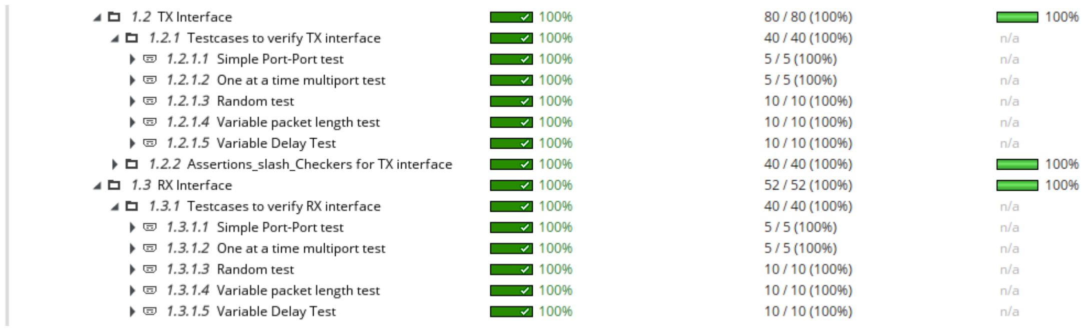
Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	==$ME()
post_session_script	htax_regress.eshaanmandal.24_12_03_17_02_05_9451	40	40	0	0	0	0	12/3/24 5:02 PM	eshaanmandal
completed	htax_regress.eshaanmandal.24_12_03_16_07_58_7996	40	39	1	0	0	0	12/3/24 4:07 PM	eshaanmandal

TestCases Mapped:



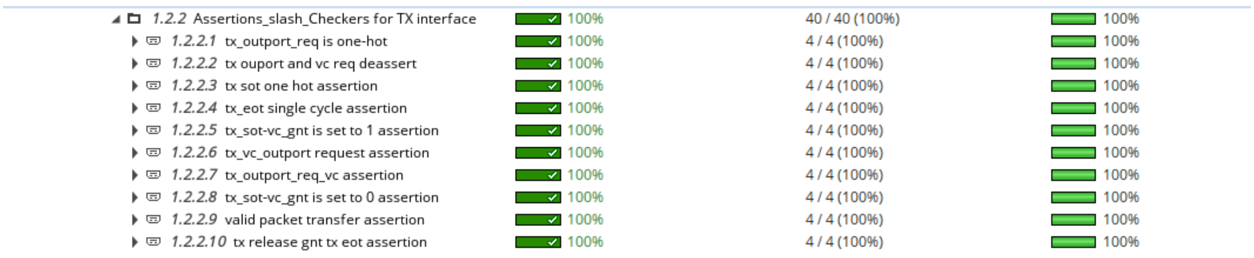
```
CSCE 616 Project Fall2024
  1 HTAX_v014
    1.1 System Interface
    1.2 TX Interface
      1.2.1 Testcases to verify TX interface
        1.2.1.1 Simple Port-Port test
        1.2.1.2 One at a time multiport test
        1.2.1.3 Random test
        1.2.1.4 Variable packet length test
        1.2.1.5 Variable Delay Test
      1.2.2 Assertions_slash_Checkers for TX interface
    1.3 RX Interface
      1.3.1 Testcases to verify RX interface
        1.3.1.1 Simple Port-Port test
        1.3.1.2 One at a time multiport test
        1.3.1.3 Random test
        1.3.1.4 Variable packet length test
        1.3.1.5 Variable Delay Test
      1.3.2 Assertions_slash_Checkers for RX interface
    1.4 Burst Mode
    1.5 HTOC Protocol
    1.6 Functional Coverage
    1.7 Code Coverage
```

```
Verification Metrics
  default
    Tests
      all_test
        simple_test_1
        simple_test_2
        simple_test_3
        simple_test_4
        simple_test_5
        simple_test_6
        random_parallel_multiport_test
        variable_delay_test
        variable_length_test
    Types
    Instances
      uvm_pkg
      htax_pkg
      top
```



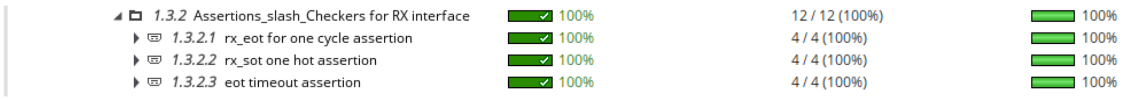
1.2 TX Interface	100%	80 / 80 (100%)	100%
1.2.1 Testcases to verify TX interface	100%	40 / 40 (100%)	n/a
1.2.1.1 Simple Port-Port test	100%	5 / 5 (100%)	n/a
1.2.1.2 One at a time multiport test	100%	5 / 5 (100%)	n/a
1.2.1.3 Random test	100%	10 / 10 (100%)	n/a
1.2.1.4 Variable packet length test	100%	10 / 10 (100%)	n/a
1.2.1.5 Variable Delay Test	100%	10 / 10 (100%)	n/a
1.2.2 Assertions_slash_Checkers for TX interface	100%	40 / 40 (100%)	100%
1.3 RX Interface	100%	52 / 52 (100%)	100%
1.3.1 Testcases to verify RX interface	100%	40 / 40 (100%)	n/a
1.3.1.1 Simple Port-Port test	100%	5 / 5 (100%)	n/a
1.3.1.2 One at a time multiport test	100%	5 / 5 (100%)	n/a
1.3.1.3 Random test	100%	10 / 10 (100%)	n/a
1.3.1.4 Variable packet length test	100%	10 / 10 (100%)	n/a
1.3.1.5 Variable Delay Test	100%	10 / 10 (100%)	n/a

Assertions/Checkers for Tx interface:



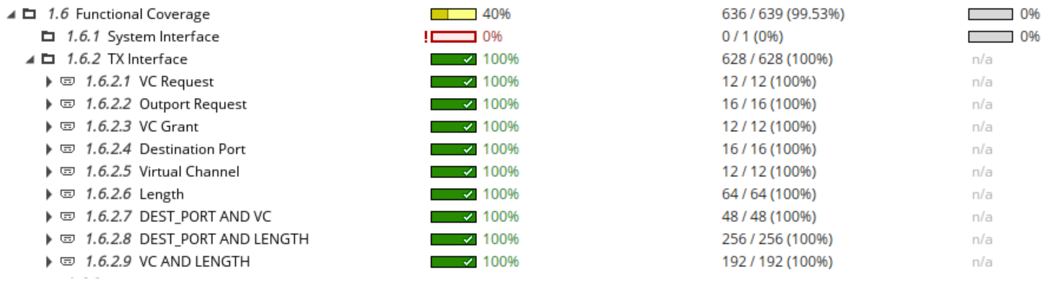
1.2.2 Assertions_slash_Checkers for TX interface	100%	40 / 40 (100%)	100%
1.2.2.1 tx_outport_req is one-hot	100%	4 / 4 (100%)	100%
1.2.2.2 tx ouport and vc req deassert	100%	4 / 4 (100%)	100%
1.2.2.3 tx sot one hot assertion	100%	4 / 4 (100%)	100%
1.2.2.4 tx_eot single cycle assertion	100%	4 / 4 (100%)	100%
1.2.2.5 tx_sot-vc_gnt is set to 1 assertion	100%	4 / 4 (100%)	100%
1.2.2.6 tx_vc_outport request assertion	100%	4 / 4 (100%)	100%
1.2.2.7 tx_outport_req_vc assertion	100%	4 / 4 (100%)	100%
1.2.2.8 tx_sot-vc_gnt is set to 0 assertion	100%	4 / 4 (100%)	100%
1.2.2.9 valid packet transfer assertion	100%	4 / 4 (100%)	100%
1.2.2.10 tx release gnt tx eot assertion	100%	4 / 4 (100%)	100%

Assertions/Checkers for Rx interface:



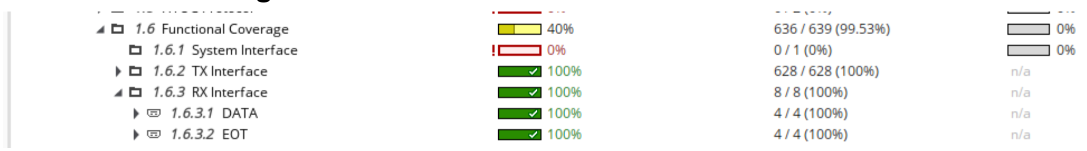
1.3.2 Assertions_slash_Checkers for RX interface	100%	12 / 12 (100%)	100%
1.3.2.1 rx_eot for one cycle assertion	100%	4 / 4 (100%)	100%
1.3.2.2 rx_sot one hot assertion	100%	4 / 4 (100%)	100%
1.3.2.3 eot timeout assertion	100%	4 / 4 (100%)	100%

Functional Coverage for Tx interface:



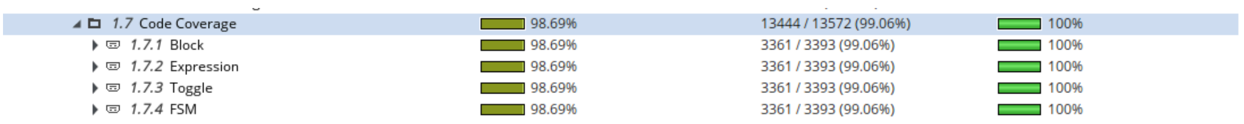
1.6 Functional Coverage	40%	636 / 639 (99.53%)	0%
1.6.1 System Interface	! 0%	0 / 1 (0%)	0%
1.6.2 TX Interface	100%	628 / 628 (100%)	n/a
1.6.2.1 VC Request	100%	12 / 12 (100%)	n/a
1.6.2.2 Outport Request	100%	16 / 16 (100%)	n/a
1.6.2.3 VC Grant	100%	12 / 12 (100%)	n/a
1.6.2.4 Destination Port	100%	16 / 16 (100%)	n/a
1.6.2.5 Virtual Channel	100%	12 / 12 (100%)	n/a
1.6.2.6 Length	100%	64 / 64 (100%)	n/a
1.6.2.7 DEST_PORT AND VC	100%	48 / 48 (100%)	n/a
1.6.2.8 DEST_PORT AND LENGTH	100%	256 / 256 (100%)	n/a
1.6.2.9 VC AND LENGTH	100%	192 / 192 (100%)	n/a

Functional Coverage for Rx interface:



1.6 Functional Coverage	40%	636 / 639 (99.53%)	0%
1.6.1 System Interface	! 0%	0 / 1 (0%)	0%
1.6.2 TX Interface	100%	628 / 628 (100%)	n/a
1.6.3 RX Interface	100%	8 / 8 (100%)	n/a
1.6.3.1 DATA	100%	4 / 4 (100%)	n/a
1.6.3.2 EOT	100%	4 / 4 (100%)	n/a

Code Coverage:



1.7 Code Coverage	98.69%	13444 / 13572 (99.06%)	100%
1.7.1 Block	98.69%	3361 / 3393 (99.06%)	100%
1.7.2 Expression	98.69%	3361 / 3393 (99.06%)	100%
1.7.3 Toggle	98.69%	3361 / 3393 (99.06%)	100%
1.7.4 FSM	98.69%	3361 / 3393 (99.06%)	100%

Code Coverage Holes:
[Explain any code coverage holes (if any)]

Bugs report

Bug 1

What is the bug?
The bug was detected when doing a parallel multiport test in the rx_interface. The assertion ***" rx_eot_timeout_check"*** failed for htax_rx_intf[3]. Implying the eot signal was not being asserted as it should have

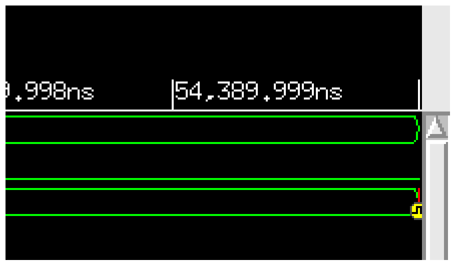
Where is it?
Module: htax_outport_data_mux (in the design directory)

File: htax_outport_data_mux.v

Line number(s): Line 43

How to reproduce:

1. After going through the Vmanager logs I was able to find the seed value which caused the assertion to fail.
2. Some of the values on this seed are as follows
    a. Seed : 396326146
    b. Delay : 6
    c. Dest_port: 3
    d. Pkt Length: 8
3. Re ran the single test (parallel multiport test) with seed value of 396326146
    xrun -f run.f +UVM_TESTNAME=multiport_parallel_random_test -svseed 396326146
4. Got the assertion failure and opened the waveform using simvision to further investigate



5. Might be tough to see but the assertion is failing at the very end.

Expected behavior:
The eot signal in the rx_intf should be asserted <= 1000 cycles.

Actual behavior:
The eot signal isnt being asserted on time, causing the assertion to fail.

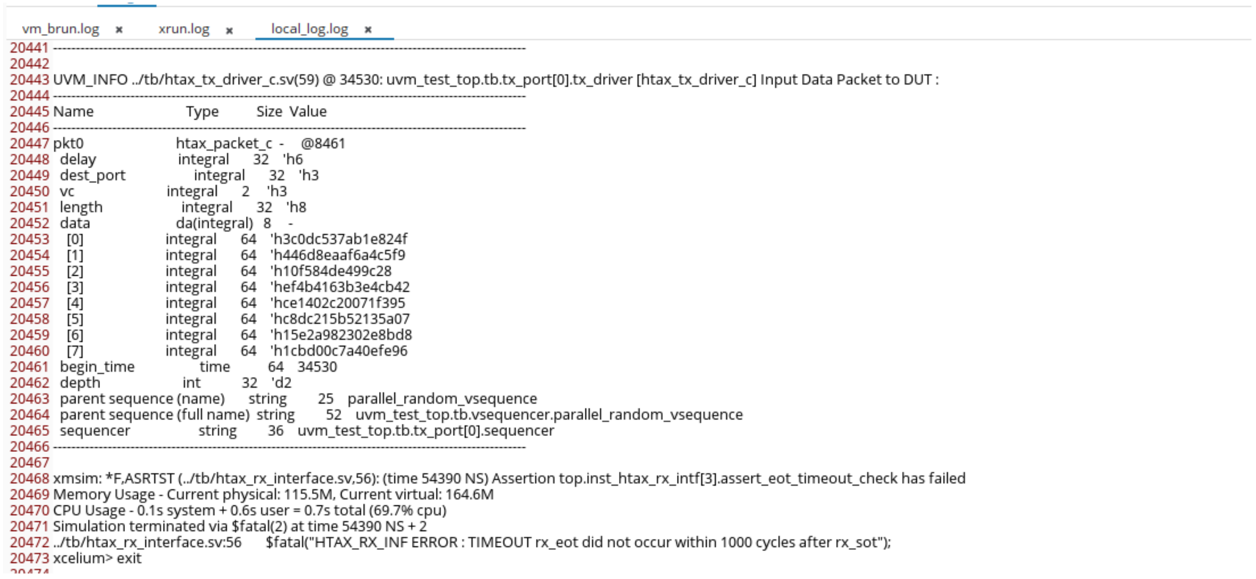
Bug fix:

On going through the DUT, the eot signal is only mentioned in htax_outport_data_mux.v. So we can narrow down our investigation to this particular file. In line number 43, we have

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

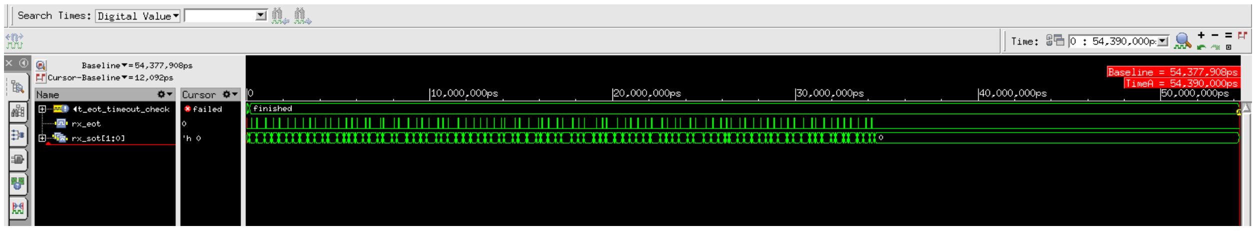
eot_in is a 4 bit signal, if all 4 bits are 1 i.e. 4'b1111. The *~(&(eot_in))* term becomes 0, which makes selected_eot = 0. Which is wrong since depending on the inport_sel_ref (if any import is selected) the selected_eot should be assigned to 1.

Failing Assertion:



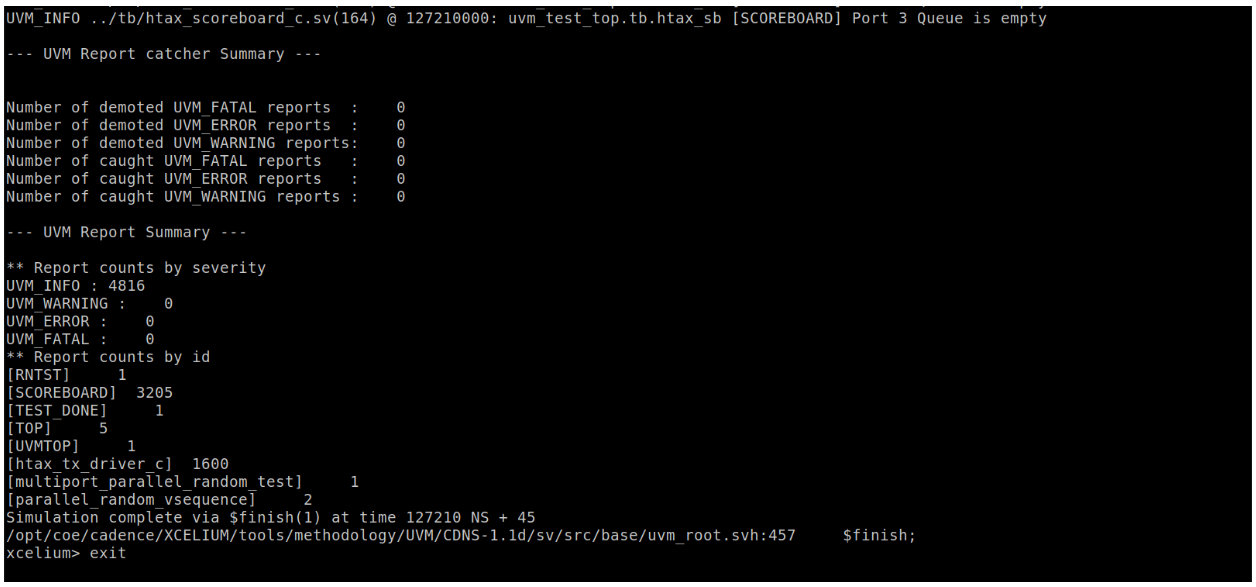
```
vm_brun.log   xrun.log   local_log.log
----------------------------------------------------------------------------------------------------

UVM_INFO ../tb/htax_tx_driver_c.sv(59) @ 34530: uvm_test_top.tb.tx_port[0].tx_driver [htax_tx_driver_c] Input Data Packet to DUT :
----------------------------------------------------------------------------------------------------
Name                     Type        Size  Value
----------------------------------------------------------------------------------------------------
pkt0                   htax_packet_c  -    @8461
  delay                  integral     32   'h6
  dest_port                integral     32   'h3
  vc                   integral     2    'h3
  length                 integral     32   'h8
  data                   da(integral)  8    -
   [0]                 integral     64   'h3c0dc537ab1e824f
   [1]                 integral     64   'h446d8eaaf6a4c5f9
   [2]                 integral     64   'h10f584de499c28
   [3]                 integral     64   'hef4b4163b3e4cb42
   [4]                 integral     64   'hce1402c20071f395
   [5]                 integral     64   'hc8dc215b52135a07
   [6]                 integral     64   'h15e2a982302e8bd8
   [7]                 integral     64   'h1cbd00c7a40efe96
  begin_time               time        64   34530
  depth                  int         32   'd2
  parent sequence (name)      string       25   parallel_random_vsequence
  parent sequence (full name)  string       52   uvm_test_top.tb.vsequencer.parallel_random_vsequence
  sequencer                string       36   uvm_test_top.tb.tx_port[0].sequencer
----------------------------------------------------------------------------------------------------

xmsim: *F,ASRTST (../tb/htax_rx_interface.sv,56): (time 54390 NS) Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed
Memory Usage - Current physical: 115.5M, Current virtual: 164.6M
CPU Usage - 0.1s system + 0.6s user = 0.7s total (69.7% cpu)
Simulation terminated via $fatal(2) at time 54390 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
```

Failing Scenario Waveform:



```
Search Times: Digital Value
Time: 0 : 54,390,000ps
Baseline = 54,377,908ps
Cursor-Baseline = 12,092ps
Baseline = 54,377,908ps
TimeA = 54,390,000ps

Name                   Cursor
t_eot_timeout_check    failed
rx_eot                 0
rx_sot[1:0]            'h 0
```

Failing Assertion Passing after the fix:



```
UVM_INFO ../tb/htax_scoreboard_c.sv(164) @ 127210000: uvm_test_top.tb.htax_sb [SCOREBOARD] Port 3 Queue is empty

--- 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 : 4816
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
[RNTST]     1
[SCOREBOARD]  3205
[TEST_DONE]     1
[TOP]     5
[UVMTOP]     1
[htax_tx_driver_c]  1600
[multiport_parallel_random_test]     1
[parallel_random_vsequence]     2
Simulation complete via $finish(1) at time 127210 NS + 45
/opt/coe/cadence/XCELIUM/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_root.svh:457     $finish;
xcelium> exit
```