

TEST CASES

Short Packet Random Test

```

class short_packet_random_test extends base_test;
  uvm_component_utils(short_packet_random_test);
  function new (string name, uvm_component parent);
    super.new(name, parent);
  endfunction : new
  function void build_phase(uvm_phase phase);
    uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", short_packet_random_vsequence::type_id::get());
    super.build_phase(phase);
  endfunction : build_phase
  task run_phase(uvm_phase phase);
    super.run_phase(phase);
    'uvm_info(get_type_name(), "Starting short packet random test", UVM_NONE)
  endtask : run_phase
endclass : short_packet_random_test

//////////////////// VIRTUAL SEQUENCE //////////////////////
class short_packet_random_vsequence extends htax_base_vseq;
  'uvm_object_utils(short_packet_random_vsequence)
  htax_packet_c req[4];
  function new (string name = "short_packet_random_vsequence");
    super.new(name);
    req[0] = new();
    req[1] = new();
    req[2] = new();
    req[3] = new();
  endfunction : new
  task body();
    repeat(500) begin
      fork
        'uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length inside {3:10}});
        'uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length inside {3:10}});
        'uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length inside {3:10}});
        'uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length inside {3:10}});
      join
    end
  endtask : body
endclass : short_packet_random_vsequence

```

Verilog file length: 1,759 lines: 48 Ln: 33 Col: 20 Pos: 1,270 Unix (LF) UTF-8 IN

Fixed Length Fixed Delay Test

```

class fixed_length_fixed_delay_test extends base_test;
  uvm_component_utils(fixed_length_fixed_delay_test);
  function new (string name, uvm_component parent);
    super.new(name, parent);
  endfunction : new
  function void build_phase(uvm_phase phase);
    uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", fixed_length_fixed_delay_vsequence::type_id::get());
    super.build_phase(phase);
  endfunction : build_phase
  task run_phase(uvm_phase phase);
    super.run_phase(phase);
    'uvm_info(get_type_name(), "Starting short packet short delay test", UVM_NONE)
  endtask : run_phase
endclass : fixed_length_fixed_delay_test

//////////////////// VIRTUAL SEQUENCE //////////////////////
class fixed_length_fixed_delay_vsequence extends htax_base_vseq;
  'uvm_object_utils(fixed_length_fixed_delay_vsequence)
  htax_packet_c req[4];
  function new (string name = "fixed_length_fixed_delay_vsequence");
    super.new(name);
    req[0] = new();
    req[1] = new();
    req[2] = new();
    req[3] = new();
  endfunction : new
  task body();
    repeat(500) begin
      fork
        'uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length == 3; req[0].delay == 4});
        'uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length == 3; req[1].delay == 4});
        'uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length == 3; req[2].delay == 4});
        'uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length == 3; req[3].delay == 4});
      join
    end
  endtask : body
endclass : fixed_length_fixed_delay_vsequence

```

Verilog file length: 1,850 lines: 49 Ln: 35 Col: 15 Pos: 1,326 Unix (LF) UTF-8 IN

Medium Packet Random Test

```
*ZAcsc616\lab-10-SaiManishGpu\lab10\test\medium_packet_random_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
medium_packet_fixed_vc_test.sv medium_packet_random_test.sv short_packet_fixed_vc_test.sv short_packet_random_test.sv short_packet_short_delay_test.sv
7 class medium_packet_random_test extends base_test;
8
9 `uvm_component_utils(medium_packet_random_test)
10 function new (string name, uvm_component parent);
11     super.new(name, parent);
12 endfunction : new
13 function void build_phase(uvm_phase phase);
14     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", medium_packet_random_vsequence::type_id::get());
15     super.build_phase(phase);
16 endfunction : build_phase
17 task run_phase(uvm_phase phase);
18     super.run_phase(phase);
19     `uvm_info(get_type_name(), "Starting medium packet random test", UVM_NONE)
20 endtask : run_phase
21
22 endclass : medium_packet_random_test
23
24 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
25
26 class medium_packet_random_vsequence extends htax_base_vseq;
27
28 `uvm_object_utils(medium_packet_random_vsequence)
29 htax_packet_c req[4];
30 function new (string name = "medium_packet_random_vsequence");
31     super.new(name);
32     req[0] = new();
33     req[1] = new();
34     req[2] = new();
35     req[3] = new();
36 endfunction : new
37 task body();
38     repeat(500) begin
39         fork
40             `uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[10:40]}});
41             `uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[10:40]}});
42             `uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[10:40]}});
43             `uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[10:40]}});
44         join
45     end
46 endtask : body
47
48 endclass : medium packet random vsequence
Verilog file length: 1,774 lines: 49 Ln: 21 Col: 5 Pos: 847 Unix (LF) UTF-8 IN
```

Long Packet Random Test

```
*Z\csce_616\lab-10-SaiManishGpu\lab10\test\long_packet_random_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
long_packet_short_delay_test.sv long_packet_random_test.sv long_packet_fixed_vc_test.sv medium_packet_fixed_vc_test.sv
7 class long_packet_random_test extends base_test;
8
9 `uvm_component_utils(long_packet_random_test)
10 function new (string name, uvm_component parent);
11     super.new(name, parent);
12 endfunction : new
13 function void build_phase(uvm_phase phase);
14     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", long_packet_random_vsequence::type_id::get());
15     super.build_phase(phase);
16 endfunction : build_phase
17 task run_phase(uvm_phase phase);
18     super.run_phase(phase);
19     `uvm_info(get_type_name(), "Starting long packet random test", UVM_NONE)
20 endtask : run_phase
21
22 endclass : long_packet_random_test
23
24 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
25
26 class long_packet_random_vsequence extends htax_base_vseq;
27
28 `uvm_object_utils(long_packet_random_vsequence)
29 rand int port;
30 htax_packet_c req[4];
31
32 function new (string name = "long_packet_random_vsequence");
33     super.new(name);
34     req[0] = new();
35     req[1] = new();
36     req[2] = new();
37     req[3] = new();
38 endfunction : new
39 task body();
40     repeat(500) begin
41         fork
42             `uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[40:50]}});
43             `uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[40:50]}});
44             `uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[40:50]}});
45             `uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[40:50]}});
46         join
47     end
48 endtask : body
Verilog file length: 1,774 lines: 50 Ln: 3 Col: 41 Pos: 141 Unix (LF) UTF-8 IN
```

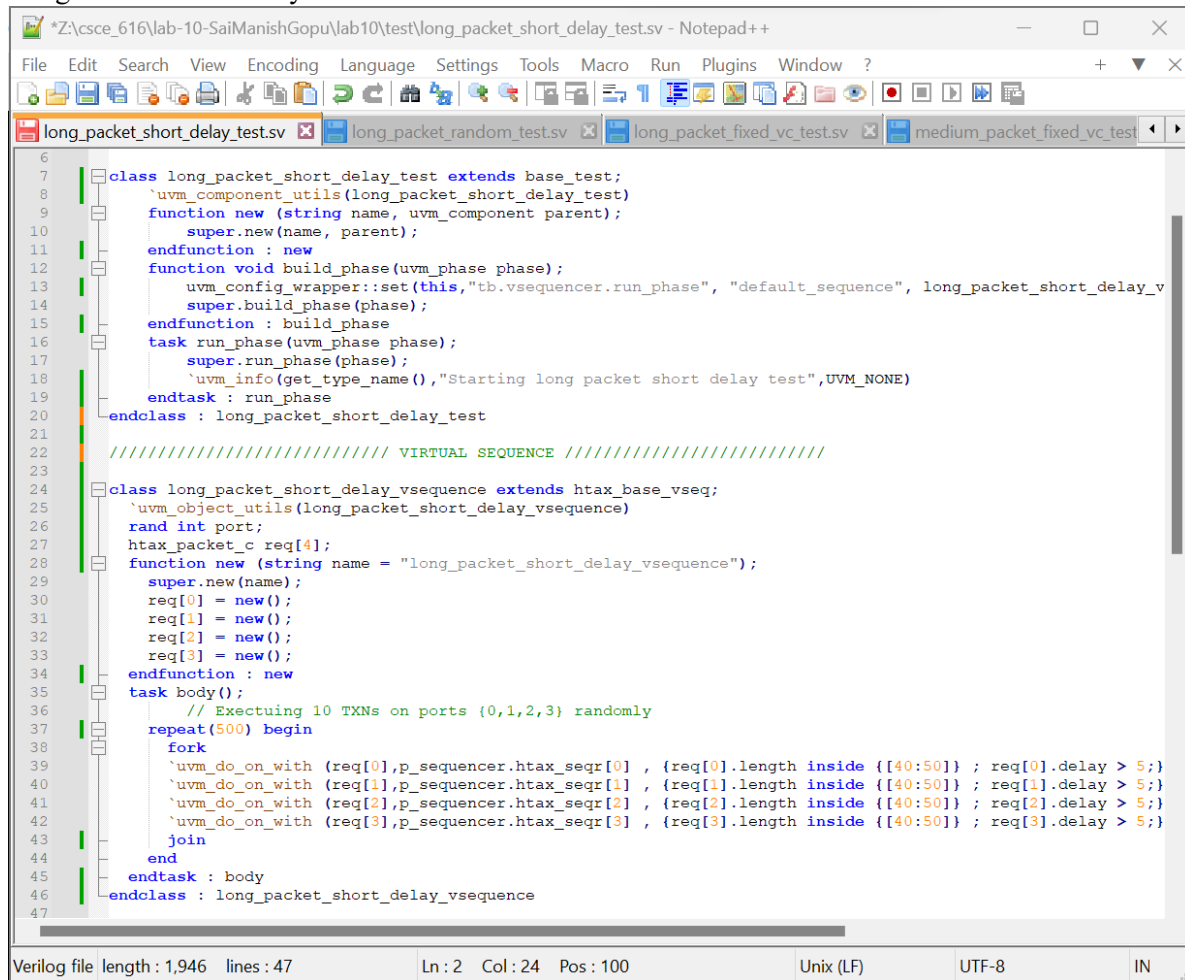
Short Packet Short Delay Test

```
*Z:\csce_616\lab-10-SaiManishGopu\lab10\test\short_packet_short_delay_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
medium_packet_fixed_vc_test.sv medium_packet_random_test.sv short_packet_short_delay_test.sv short_packet_fixed_vc_test.sv short_packet_random_test.sv
7 class short_packet_short_delay_test extends base_test;
8   uvm_component_utils(short_packet_short_delay_test)
9   function new (string name, uvm_component parent);
10     super.new(name, parent);
11   endfunction : new
12   function void build_phase(uvm_phase phase);
13     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", short_packet_short_delay_vsequence::type_id::get());
14     super.build_phase(phase);
15   endfunction : build_phase
16   task run_phase(uvm_phase phase);
17     super.run_phase(phase);
18     `uvm_info(get_type_name(), "Starting short packet short delay test", UVM_NONE)
19   endtask : run_phase
20 endclass : short_packet_short_delay_test
21
22 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
23
24 class short_packet_short_delay_vsequence extends htax_base_vseq;
25   `uvm_object_utils(short_packet_short_delay_vsequence)
26   htax_packet_c req[4];
27   function new (string name = "short_packet_short_delay_vsequence");
28     super.new(name);
29     req[0] = new();
30     req[1] = new();
31     req[2] = new();
32     req[3] = new();
33   endfunction : new
34
35   task body();
36     repeat(500) begin
37
38       fork
39         `uvm_do_on_with (req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[3:10]}; req[0].delay > 5;});
40         `uvm_do_on_with (req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[3:10]}; req[1].delay > 5;});
41         `uvm_do_on_with (req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[3:10]}; req[2].delay > 5;});
42         `uvm_do_on_with (req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[3:10]}; req[3].delay > 5;});
43       join
44
45     end
46   endtask : body
47
48 endclass : short packet short delay vsequence
Verilog file length: 1,893 lines: 49 Ln: 43 Col: 11 Pos: 1,818 Unix (LF) UTF-8 IN
```

Medium Packet Short Delay Test

```
*Z:\csce_616\lab-10-SaiManishGopu\lab10\test\medium_packet_short_delay_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
medium_packet_fixed_vc_test.sv medium_packet_random_test.sv medium_packet_short_delay_test.sv short_packet_fixed_vc_test.sv short_packet_random_test.sv
6 class medium_packet_short_delay_test extends base_test;
7   uvm_component_utils(medium_packet_short_delay_test)
8   function new (string name, uvm_component parent);
9     super.new(name, parent);
10   endfunction : new
11   function void build_phase(uvm_phase phase);
12     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", medium_packet_short_delay_vsequence::type_id::get());
13     super.build_phase(phase);
14   endfunction : build_phase
15   task run_phase(uvm_phase phase);
16     super.run_phase(phase);
17     `uvm_info(get_type_name(), "Starting medium packet short delay test", UVM_NONE)
18   endtask : run_phase
19 endclass : medium_packet_short_delay_test
20
21 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
22
23 class medium_packet_short_delay_vsequence extends htax_base_vseq;
24   `uvm_object_utils(medium_packet_short_delay_vsequence)
25   htax_packet_c req[4];
26   function new (string name = "medium_packet_short_delay_vsequence");
27     super.new(name);
28     req[0] = new();
29     req[1] = new();
30     req[2] = new();
31     req[3] = new();
32   endfunction : new
33
34   task body();
35     repeat(500) begin
36
37       fork
38         `uvm_do_on_with (req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[10:40]}; req[0].delay > 5;});
39         `uvm_do_on_with (req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[10:40]}; req[1].delay > 5;});
40         `uvm_do_on_with (req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[10:40]}; req[2].delay > 5;});
41         `uvm_do_on_with (req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[10:40]}; req[3].delay > 5;});
42       join
43
44     end
45   endtask : body
46
47 endclass : medium_packet_short_delay_vsequence
Verilog file length: 1,890 lines: 45 Ln: 43 Col: 17 Pos: 1,843 Unix (LF) UTF-8 IN
```

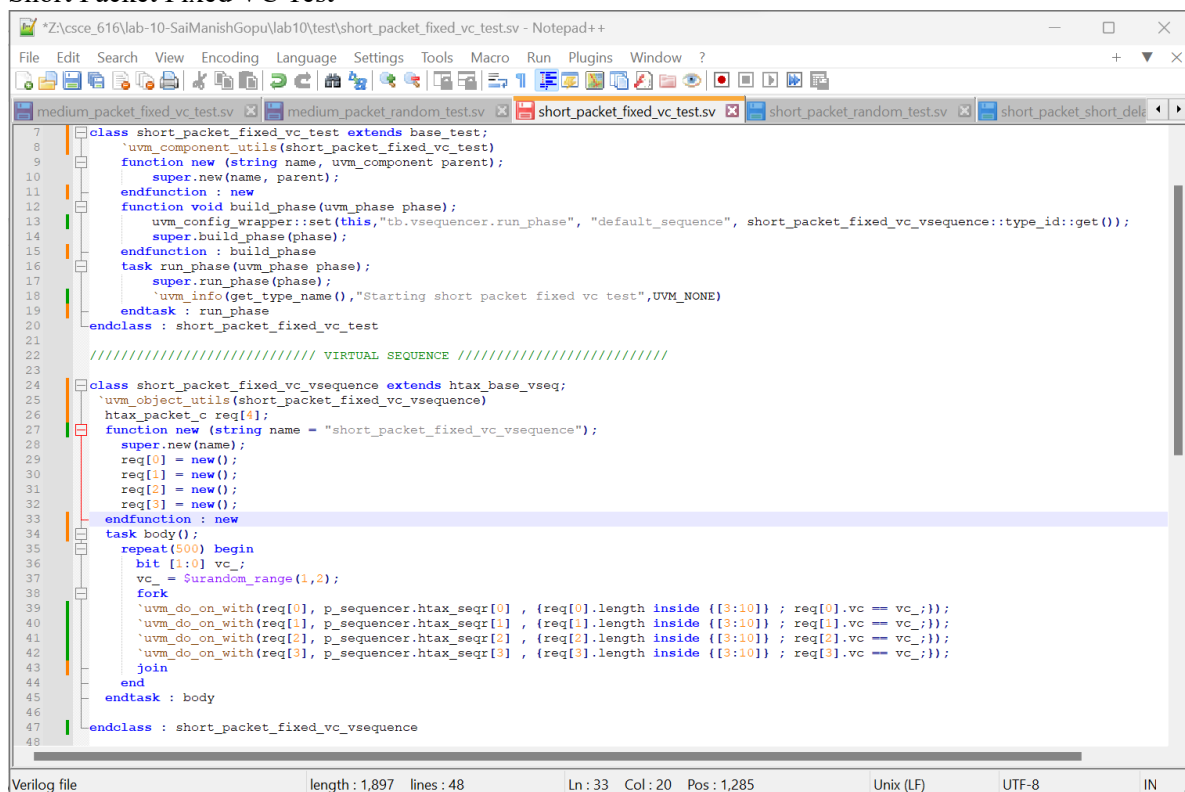
Long Packet Short Delay Test



```
6
7 class long_packet_short_delay_test extends base_test;
8   `uvm_component_utils(long_packet_short_delay_test)
9   function new (string name, uvm_component parent);
10     super.new(name, parent);
11   endfunction : new
12   function void build_phase(uvm_phase phase);
13     uvm_config_wrapper::set(this,"tb.vsequencer.run_phase", "default_sequence", long_packet_short_delay_v
14     super.build_phase(phase);
15   endfunction : build_phase
16   task run_phase(uvm_phase phase);
17     super.run_phase(phase);
18     `uvm_info(get_type_name(),"Starting long packet short delay test",UVM_NONE)
19   endtask : run_phase
20 endclass : long_packet_short_delay_test
21
22 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
23
24 class long_packet_short_delay_vsequence extends htax_base_vseq;
25   `uvm_object_utils(long_packet_short_delay_vsequence)
26   rand int port;
27   htax_packet_c req[4];
28   function new (string name = "long_packet_short_delay_vsequence");
29     super.new(name);
30     req[0] = new();
31     req[1] = new();
32     req[2] = new();
33     req[3] = new();
34   endfunction : new
35   task body();
36     // Exectuing 10 TXNs on ports {0,1,2,3} randomly
37     repeat(500) begin
38       fork
39         `uvm_do_on_with (req[0],p_sequencer.htax_seqr[0] , {req[0].length inside {[40:50]} ; req[0].delay > 5;}
40         `uvm_do_on_with (req[1],p_sequencer.htax_seqr[1] , {req[1].length inside {[40:50]} ; req[1].delay > 5;}
41         `uvm_do_on_with (req[2],p_sequencer.htax_seqr[2] , {req[2].length inside {[40:50]} ; req[2].delay > 5;}
42         `uvm_do_on_with (req[3],p_sequencer.htax_seqr[3] , {req[3].length inside {[40:50]} ; req[3].delay > 5;}
43       join
44     end
45   endtask : body
46 endclass : long_packet_short_delay_vsequence
47
```

Verilog file length : 1,946 lines : 47 Ln : 2 Col : 24 Pos : 100 Unix (LF) UTF-8 IN

Short Packet Fixed VC Test



```
7
8 class short_packet_fixed_vc_test extends base_test;
9   `uvm_component_utils(short_packet_fixed_vc_test)
10   function new (string name, uvm_component parent);
11     super.new(name, parent);
12   endfunction : new
13   function void build_phase(uvm_phase phase);
14     uvm_config_wrapper::set(this,"tb.vsequencer.run_phase", "default_sequence", short_packet_fixed_vc_vsequence::type_id::get());
15     super.build_phase(phase);
16   endfunction : build_phase
17   task run_phase(uvm_phase phase);
18     super.run_phase(phase);
19     `uvm_info(get_type_name(),"Starting short packet fixed vc test",UVM_NONE)
20   endtask : run_phase
21 endclass : short_packet_fixed_vc_test
22
23 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
24
25 class short_packet_fixed_vc_vsequence extends htax_base_vseq;
26   `uvm_object_utils(short_packet_fixed_vc_vsequence)
27   htax_packet_c req[4];
28   function new (string name = "short_packet_fixed_vc_vsequence");
29     super.new(name);
30     req[0] = new();
31     req[1] = new();
32     req[2] = new();
33     req[3] = new();
34   endfunction : new
35   task body();
36     repeat(500) begin
37       bit [1:0] vc_;
38       vc_ = $urandom_range(1,2);
39       fork
40         `uvm_do_on_with(req[0], p_sequencer.htax_seqr[0] , {req[0].length inside {[3:10]} ; req[0].vc == vc_;});
41         `uvm_do_on_with(req[1], p_sequencer.htax_seqr[1] , {req[1].length inside {[3:10]} ; req[1].vc == vc_;});
42         `uvm_do_on_with(req[2], p_sequencer.htax_seqr[2] , {req[2].length inside {[3:10]} ; req[2].vc == vc_;});
43         `uvm_do_on_with(req[3], p_sequencer.htax_seqr[3] , {req[3].length inside {[3:10]} ; req[3].vc == vc_;});
44       join
45     end
46   endtask : body
47 endclass : short_packet_fixed_vc_vsequence
48
```

Verilog file length : 1,897 lines : 48 Ln : 33 Col : 20 Pos : 1,285 Unix (LF) UTF-8 IN

Medium Packet Fixed VC Test

```
*Z:\csce_616\lab-10-SaiManishGopu\lab10\test\medium_packet_fixed_vc_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
medium_packet_fixed_vc_test.sv medium_packet_random_test.sv short_packet_fixed_vc_test.sv short_packet_random_test.sv short_packet_short_delay_test.sv
7 class medium_packet_fixed_vc_test extends base_test;
8   `uvm_component_utils(medium_packet_fixed_vc_test)
9   function new (string name, uvm_component parent);
10     super.new(name, parent);
11   endfunction : new
12   function void build_phase(uvm_phase phase);
13     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", medium_packet_fixed_vc_vsequence::type_id::get());
14     super.build_phase(phase);
15   endfunction : build_phase
16   task run_phase(uvm_phase phase);
17     super.run_phase(phase);
18     `uvm_info(get_type_name(), "Starting medium packet fixed vc test", UVM_NONE)
19   endtask : run_phase
20 endclass : medium_packet_fixed_vc_test
21
22 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
23
24 class medium_packet_fixed_vc_vsequence extends htax_base_vseq;
25   `uvm_object_utils(medium_packet_fixed_vc_vsequence)
26   htax_packet_c req[4];
27   function new (string name = "medium_packet_fixed_vc_vsequence");
28     super.new(name);
29     req[0] = new();
30     req[1] = new();
31     req[2] = new();
32     req[3] = new();
33   endfunction : new
34
35   task body();
36     repeat(500) begin
37       bit [1:0] vc_;
38       vc_ = $urandom_range(1,2);
39       fork
40         `uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[10:40]}; req[0].vc == vc_});
41         `uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[10:40]}; req[1].vc == vc_});
42         `uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[10:40]}; req[2].vc == vc_});
43         `uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[10:40]}; req[3].vc == vc_});
44       join
45     end
46   endtask : body
47
48 endclass : medium_packet_fixed_vc_vsequence
Verilog file length: 1,912 lines: 49 Ln: 47 Col: 3 Pos: 1,868 Unix (LF) UTF-8 IN
```

Long Packet Fixed VC Test

```
*Z:\csce_616\lab-10-SaiManishGopu\lab10\test\long_packet_fixed_vc_test.sv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
long_packet_short_delay_test.sv long_packet_random_test.sv long_packet_fixed_vc_test.sv medium_packet_fixed_vc_test.sv
6
7 class long_packet_fixed_vc_test extends base_test;
8   `uvm_component_utils(long_packet_fixed_vc_test)
9   function new (string name, uvm_component parent);
10     super.new(name, parent);
11   endfunction : new
12   function void build_phase(uvm_phase phase);
13     uvm_config_wrapper::set(this, "tb.vsequencer.run_phase", "default_sequence", long_packet_fixed_vc_vsequence::type_id::get());
14     super.build_phase(phase);
15   endfunction : build_phase
16   task run_phase(uvm_phase phase);
17     super.run_phase(phase);
18     `uvm_info(get_type_name(), "Starting long packet fixed vc test", UVM_NONE)
19   endtask : run_phase
20 endclass : long_packet_fixed_vc_test
21
22 ////////////////////////////////////////////////// VIRTUAL SEQUENCE ///////////////////////////////////
23
24 class long_packet_fixed_vc_vsequence extends htax_base_vseq;
25   `uvm_object_utils(long_packet_fixed_vc_vsequence)
26   htax_packet_c req[4];
27   function new (string name = "long_packet_fixed_vc_vsequence");
28     super.new(name);
29     req[0] = new();
30     req[1] = new();
31     req[2] = new();
32     req[3] = new();
33   endfunction : new
34
35   task body();
36     bit [1:0] vc_;
37     vc_ = $urandom_range(1,2);
38     fork
39       `uvm_do_on_with(req[0], p_sequencer.htax_seqr[0], {req[0].length inside {[40:50]}; req[0].vc == vc_});
40       `uvm_do_on_with(req[1], p_sequencer.htax_seqr[1], {req[1].length inside {[40:50]}; req[1].vc == vc_});
41       `uvm_do_on_with(req[2], p_sequencer.htax_seqr[2], {req[2].length inside {[40:50]}; req[2].vc == vc_});
42       `uvm_do_on_with(req[3], p_sequencer.htax_seqr[3], {req[3].length inside {[40:50]}; req[3].vc == vc_});
43     join
44   endtask : body
45
46 endclass : long_packet_fixed_vc_vsequence
Verilog file length: 1,875 lines: 47 Ln: 41 Col: 112 Pos: 1,793 Unix (LF) UTF-8 IN
```

BUG REPORT

Regression Summary

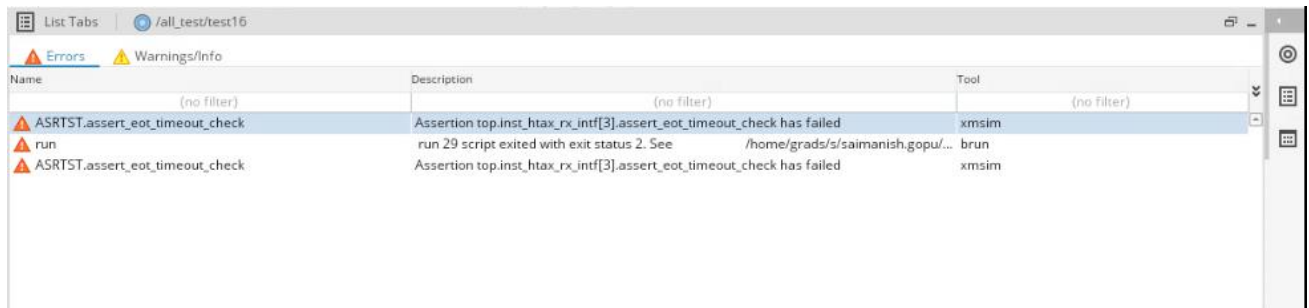
vManager hns3.ace.tamu.edu:8081 646 saimanish.gopu (Regression Center)										
Regression Analysis Planning Tracking										
Views Global Operations Sessions Analyze Report Help										
Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner	
completed	htax_regress.saimanish.gopu.23_12_05_19_53_17_5188	119	97	22	0	0	0	12/5/23 7:53 PM	saimanish.gopu	

vPlan Hierarchy				
CSCE 616 Project Fall2022				
Ex	Unit	Name	Overall Average Grade	Assertion Status Grade
(no filter)			(no filter)	(no filter)
1	CSCE 616 Project Fall2022		99.68%	368 / 412 (89.32%)
1	HTAX_v014		99.68%	368 / 412 (89.32%)
1	1.1 TX Interface		99.08%	104 / 126 (82.54%)
1	1.1.1 Testcases to verify TX interface		98.17%	97 / 119 (81.51%)
1	1.1.1.1 simple random test		100%	5 / 5 (100%)
1	1.1.1.2 multipoint sequential random test		100%	5 / 5 (100%)
1	1.1.1.3 short packet random test		100%	1 / 1 (100%)
1	1.1.1.4 medium packet random test		100%	1 / 1 (100%)
1	1.1.1.5 long packet random test		100%	1 / 1 (100%)
1	1.1.1.6 short packet short delay test		100%	1 / 1 (100%)
1	1.1.1.7 medium packet short delay test		100%	1 / 1 (100%)
1	1.1.1.8 long packet short delay test		100%	1 / 1 (100%)
1	1.1.1.9 short packet fixed vc test		100%	1 / 1 (100%)
1	1.1.1.10 medium packet fixed vc test		100%	1 / 1 (100%)
1	1.1.1.11 long packet fixed vc test		100%	1 / 1 (100%)
1	1.1.1.12 fixed length fixed delay		78%	78 / 100 (78%)
1	1.1.2 Assertions_slash_Checkers for TX interface		100%	7 / 7 (100%)
1	1.2 RX Interface		100%	5 / 5 (100%)
1	1.2.1 Testcases to verify RX interface		100%	3 / 3 (100%)
1	1.2.2 Assertions_slash_Checkers for RX interface		100%	2 / 2 (100%)
1	1.3 Functional Coverage		100%	113 / 113 (100%)
1	1.4 Code Coverage		99.66%	146 / 168 (86.9%)

Xrun.log

Showing 3 items	
Details	/all_test/test16
Attributes Logs	
vm_brun.log	xrun.log
938	
939	
940 xmsim: *F.ASRTST (. /tb/htax_rx_interface.sv.56): (time 21110 NS) Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed	
941 Memory Usage - Current physical: 94.4M, Current virtual: 179.9M	
942 CPU Usage - 0.1s system + 0.2s user = 0.3s total (48.0% cpu)	
943 Simulation terminated via \$fatal(2) at time 21110 NS + 2	
944 ./tb/htax_rx_interface.sv.56 \$fatal("HTAX_RX_INF ERROR: TIMEOUT rx_eot did not occur within 1000 cycles after rx_sot")	
945 xcelium> exit	
946	
947 coverage setup:	
948 workdir : /home/grads/s/saimanish.gopu/csce_616/lab-10-SaiManishGopu/lab10/sim/regression/htax_regress.saimanish.gopu.23_12_05_19_53_17_5188/chain_0/all_test/run_29/cov	
949 dutinst : top(top)	
950 scope : scope	
951 testname: test_sv-961847974	
952	
953 coverage files:	
954 model(design data) : /home/grads/s/saimanish.gopu/csce_616/lab-10-SaiManishGopu/lab10/sim/regression/htax_regress.saimanish.gopu.23_12_05_19_53_17_5188/model_dir/icc_4e1	
955 data : /home/grads/s/saimanish.gopu/csce_616/lab-10-SaiManishGopu/lab10/sim/regression/htax_regress.saimanish.gopu.23_12_05_19_53_17_5188/chain_0/all_test/run_29/c	
956 TOOL: xrun 22.03-s004: Exiting on Dec 05, 2023 at 19:58:59 CST (total: 00:00:01)	
957 Wrote vManager metrics information to /home/grads/s/saimanish.gopu/csce_616/lab-10-SaiManishGopu/lab10/sim/regression/htax_regress.saimanish.gopu.23_12_05_19_53_17_518	
958	
959 Intermediate vsf file located at /home/grads/s/saimanish.gopu/csce_616/lab-10-SaiManishGopu/lab10/sim/regression/htax_regress.saimanish.gopu.23_12_05_19_53_17_5188/chain_0	
1	
Match Case	

Error Message



Name	Description	Tool
ASRTST.assert_eot_timeout_check	Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed	xmsim
run	run 29 script exited with exit status 2. See /home/grads/s/saimanish.gopu/...	brun
ASRTST.assert_eot_timeout_check	Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed	xmsim

Debug Process

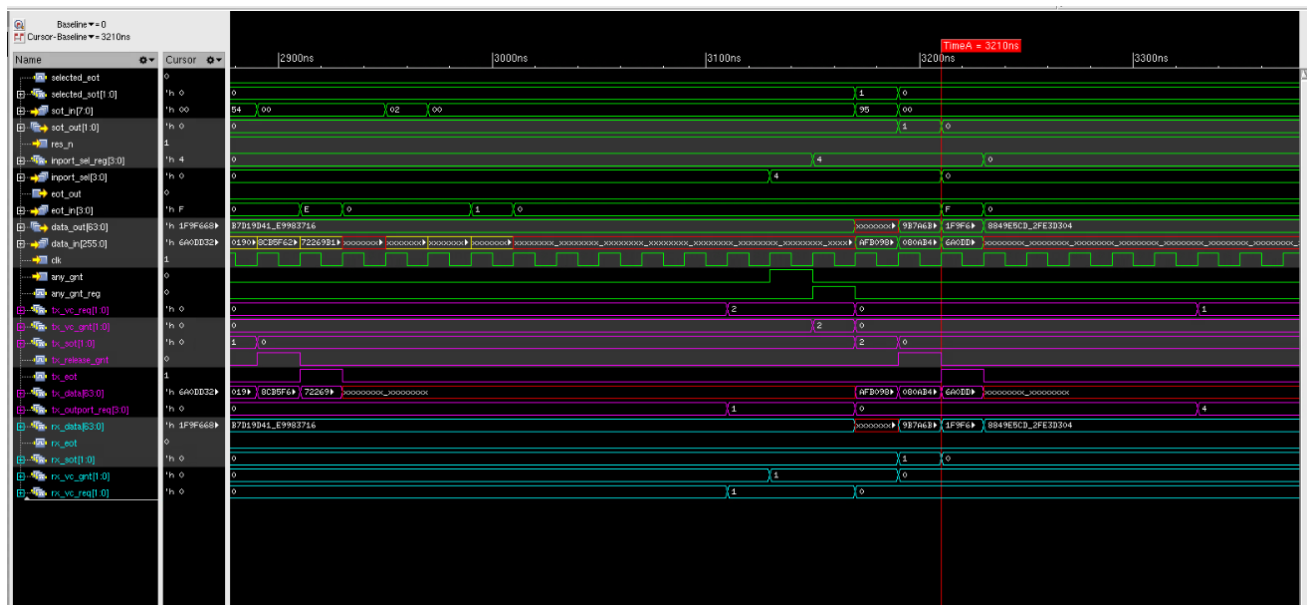
I ran multiple regressions with random seed number. Finally for the seed number 961847974 the test case fixed_length_fixed_delay_test failed and all other tests passed.

When the test case fixed_length_fixed_delay_test is ran isolated with the same seed number. It gave below error message. It says that the assertion assert_eot_timeout_check has failed.

```
xmsim: *F,ASRTST ( ../tb/htax_rx_interface.sv,56): (time 23210 NS) Assertion top.inst_htax_rx_intf[3].assert_eot_timeout_check has failed
Memory Usage - Current physical: 92.6M, Current virtual: 175.9M
CPU Usage - 0.1s system + 0.2s user = 0.3s total (67.0% cpu)
Simulation terminated via $fatal(2) at time 23210 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_sv961847974
```

To analyse this, I opened the simvision waveform which came out as below,



In the above waveform,

The signal rx_eot has not risen eventhough the transaction is tx_eot is raised which indicates the end of transaction. This could be the reason for the assertion failure. When traced, it is found that the tx_eot signal value comes from signal eot_out from the htax_output_mux_data from design folder. When the

htax_outport_mux_data is analysed, I came to know that the eot_out is coming from another signal selected_eot and selected_eot signal is assigned the value as selected_eot = |(eot_in & inport_sel_reg) & ~(eot_in)); if the if condition is satisfied i.e., true. But ideally this should not be the case.

It appears that the RX side receives the rx_sot signal, and data becomes visible, but the rx_eot signal fails to appear. The absence of rx_eot is attributed to the added term &~(&(eot_in)) in the assignment of selected_eot. This issue arises when all RX ports simultaneously request a transaction, and the packet length and delay are identical. In this scenario, all bits in eot_in[3:0] are expected to be high (indicated as 'F' in the waveform). However, as eot_in[3:0] goes high, ~(eot_in)) goes low. Consequently, when this low value is combined with other terms through bitwise AND, the result is zero.

The signals eot_out and selected_eot from the htax_outport_mux_data design file is as below

```
// Muxes for data, eot, sot
`ifndef ASYNC_RES
always @(posedge clk or negedge res_n) `else
always @(posedge clk) `endif
begin
    if (!res_n)
    begin
        data_out <= {WIDTH {1'b0}};
        sot_out <= {VC {1'b0}};
        eot_out <= 1'b0;
    end
    else
    begin
        (* full_case *) (* parallel_case *)
        casex (inport_sel_reg)
            4'b1xxx: data_out <= data_in[((4*WIDTH)-1):(3*WIDTH)];
            4'b0xxx: data_out <= data_in[((3*WIDTH)-1):(2*WIDTH)];
            4'bxx1x: data_out <= data_in[((2*WIDTH)-1):(1*WIDTH)];
            4'bxxxx1: data_out <= data_in[((1*WIDTH)-1):(0*WIDTH)];
        endcase
        if ((inport_sel && !any_gnt) )
            sot_out <= selected_sot;
        else
            sot_out <= {VC{1'b0}};
        if ((inport_sel_reg && !eot_out || (selected_sot && (inport_sel && !any_gnt) )))
            eot_out <= selected_eot;
        else
            eot_out <= 1'b0;
        end
    end
endmodule

reg                any_gnt_reg;
reg [NUM_PORTS-1:0] inport_sel_reg;
reg [VC-1:0]selected_sot;
wire               selected_eot;

always @( * )
begin
    (* full_case *) (* parallel_case *)
    casex (inport_sel)
        4'b1xxx: selected_sot = sot_in[((4*VC)-1):(3*VC)];
        4'b0xxx: selected_sot = sot_in[((3*VC)-1):(2*VC)];
        4'bxx1x: selected_sot = sot_in[((2*VC)-1):(1*VC)];
        4'bxxxx1: selected_sot = sot_in[((1*VC)-1):(0*VC)];
    endcase
end

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

`ifndef ASYNC_RES
always @(posedge clk or negedge res_n) `else
always @(posedge clk) `endif
begin
    if (!res_n) begin
        inport_sel_reg <= {NUM_PORTS{1'b0}};
        any_gnt_reg <= 0;
    end else begin
        any_gnt_reg <= any_gnt;

        //if(any_gnt_reg)
        inport_sel_reg <= inport_sel;
    end
end
```

Recommended Fix for the Bug:

The bug can be fixed by making modification to the selected_eot assignment statement by removing ~(&(eot_in)).

Therefore, the correct logic for selected_eot is

assign selected_eot = |(eot_in & inport_sel_reg);

COVERAGE REPORT

Test Coverage

vPlan Hierarchy				
CSCE 616 Project Fall2022				
Ex	Unit	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		CSCE 616 Project Fall2022	99.68%	368 / 412 (89.32%)
		HTAX_v014	99.68%	368 / 412 (89.32%)
		1.1 TX Interface	99.08%	104 / 126 (82.54%)
		1.1.1 Testcases to verify TX interface	98.17%	97 / 119 (81.51%)
		1.1.1.1 simple random test	100%	5 / 5 (100%)
		1.1.1.2 multiprot sequential random test	100%	5 / 5 (100%)
		1.1.1.3 short packet random test	100%	1 / 1 (100%)
		1.1.1.4 medium packet random test	100%	1 / 1 (100%)
		1.1.1.5 long packet random test	100%	1 / 1 (100%)
		1.1.1.6 short packet short delay test	100%	1 / 1 (100%)
		1.1.1.7 medium packet short delay test	100%	1 / 1 (100%)
		1.1.1.8 long packet short delay test	100%	1 / 1 (100%)
		1.1.1.9 short packet fixed vc test	100%	1 / 1 (100%)
		1.1.1.10 medium packet fixed vc test	100%	1 / 1 (100%)
		1.1.1.11 long packet fixed vc test	100%	1 / 1 (100%)
		1.1.1.12 fixed length fixed delay	78%	78 / 100 (78%)
		1.1.2 Assertions_slash_Checkers for TX interfacc	100%	7 / 7 (100%)
		1.2 RX Interface	100%	5 / 5 (100%)
		1.2.1 Testcases to verify RX interface	100%	3 / 3 (100%)
		1.2.2 Assertions_slash_Checkers for RX interfacc	100%	2 / 2 (100%)
		1.3 Functional Coverage	100%	113 / 113 (100%)
		1.4 Code Coverage	99.66%	146 / 168 (86.9%)

Assertion Coverage of TX and RX:

vPlan Hierarchy				
CSCE 616 Project Fall2022				
Ex	Unit	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		CSCE 616 Project Fall2022	99.68%	368 / 412 (89.32%)
		HTAX_v014	99.68%	368 / 412 (89.32%)
		1.1 TX Interface	99.08%	104 / 126 (82.54%)
		1.1.1 Testcases to verify TX interface	98.17%	97 / 119 (81.51%)
		1.1.2 Assertions_slash_Checkers for TX interfacc	100%	7 / 7 (100%)
		1.1.2.1 tx_outport_req is one-hot	100%	1 / 1 (100%)
		1.1.2.2 no tx_outport_req without tx_vc_req	100%	1 / 1 (100%)
		1.1.2.3 no tx_vc_req without tx_outport_req	100%	1 / 1 (100%)
		1.1.2.4 no tx_sot without tx_vc_gnt	100%	1 / 1 (100%)
		1.1.2.5 tx_eot for single cycle	100%	1 / 1 (100%)
		1.1.2.6 tx_release_gnt before one cycle of tx	100%	1 / 1 (100%)
		1.1.2.7 no tx_vc_sot without tx_vc_gnt_1	100%	1 / 1 (100%)
		1.2 RX Interface	100%	5 / 5 (100%)
		1.2.1 Testcases to verify RX interface	100%	3 / 3 (100%)
		1.2.2 Assertions_slash_Checkers for RX interfacc	100%	2 / 2 (100%)
		1.2.2.1 rx_sot is one hot	100%	1 / 1 (100%)
		1.2.2.2 rx_eot timeout check	100%	1 / 1 (100%)
		1.3 Functional Coverage	100%	113 / 113 (100%)
		1.4 Code Coverage	99.66%	146 / 168 (86.9%)

Code Coverage

vPlan Hierarchy				
CSCE 616 Project Fall2022				
Ex	Unit	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		CSCE 616 Project Fall2022	99.68%	368 / 412 (89.32%)
		HTAX_v014	99.68%	368 / 412 (89.32%)
		1.1 TX Interface	99.08%	104 / 126 (82.54%)
		1.2 RX Interface	100%	5 / 5 (100%)
		1.3 Functional Coverage	100%	113 / 113 (100%)
		1.4 Code Coverage	99.66%	146 / 168 (86.9%)
		1.4.1 Block	100%	7 / 7 (100%)
		1.4.2 Expression	100%	6 / 6 (100%)
		1.4.3 Toggle	100%	36 / 36 (100%)
		1.4.4 FSM	98.62%	97 / 119 (81.51%)

Functional Coverage

UNB	Name	Overall Average Grade	Overall Covered	Assertion Status Grade
	(no filter)	(no filter)	(no filter)	(no filter)
CSCE 616 Project Fall2022		99.68%	368 / 412 (89.32%)	88.89%
1	HTAX_v014	99.68%	368 / 412 (89.32%)	88.89%
1.1	TX Interface	99.08%	104 / 126 (82.54%)	100%
1.2	RX Interface	100%	5 / 5 (100%)	50%
1.3	Functional Coverage	100%	113 / 113 (100%)	n/a
1.3.1	TX Interface	100%	109 / 109 (100%)	n/a
1.3.1.1	In port	100%	4 / 4 (100%)	n/a
1.3.1.2	Outport Request	100%	4 / 4 (100%)	n/a
1.3.1.3	Length	100%	16 / 16 (100%)	n/a
1.3.1.4	length and dest_port	100%	64 / 64 (100%)	n/a
1.3.1.5	VC	100%	3 / 3 (100%)	n/a
1.3.1.6	dest port and vc	100%	12 / 12 (100%)	n/a
1.3.1.7	vc_req	100%	3 / 3 (100%)	n/a
1.3.1.8	vc_gnt	100%	3 / 3 (100%)	n/a
1.3.2	RX Interface	100%	4 / 4 (100%)	n/a
1.3.2.1	outport_req	100%	4 / 4 (100%)	n/a
1.4	Code Coverage	99.66%	146 / 168 (86.9%)	n/a

Length

UNB	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
	length[0]	100%	156773
	length[1]	100%	2963
	length[2]	100%	2653
	length[3]	100%	817
	length[4]	100%	749
	length[5]	100%	753
	length[6]	100%	799
	length[7]	100%	793
	length[8]	100%	780
	length[9]	100%	809
	length[10]	100%	2407
	length[11]	100%	2179
	length[12]	100%	1627
	length[13]	100%	9
	length[14]	100%	8
	length[15]	100%	6

Showing 16 items

VC

UNB	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
	auto[1]	100%	14659
	auto[2]	100%	14659
	auto[3]	100%	14192

Showing 3 items

Details VC

dest_port

List Tabs In port			
UNI	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
	dest_port[0]	100%	10833
	dest_port[1]	100%	10928
	dest_port[2]	100%	10876
	dest_port[3]	100%	10873

Showing 4 items

dest_port and vc

List Tabs dest port and vc				
UNI	Name	DEST_PORT	VC	Overall Average Grade
	(no filter)	(no filter)	(no filter)	(no filter)
	dest_port[0],auto[1]	dest_port[0]	auto[1]	100%
	dest_port[0],auto[2]	dest_port[0]	auto[2]	100%
	dest_port[0],auto[3]	dest_port[0]	auto[3]	100%
	dest_port[1],auto[1]	dest_port[1]	auto[1]	100%
	dest_port[1],auto[2]	dest_port[1]	auto[2]	100%
	dest_port[1],auto[3]	dest_port[1]	auto[3]	100%
	dest_port[2],auto[1]	dest_port[2]	auto[1]	100%
	dest_port[2],auto[2]	dest_port[2]	auto[2]	100%
	dest_port[2],auto[3]	dest_port[2]	auto[3]	100%
	dest_port[3],auto[1]	dest_port[3]	auto[1]	100%
	dest_port[3],auto[2]	dest_port[3]	auto[2]	100%
	dest_port[3],auto[3]	dest_port[3]	auto[3]	100%

Showing 12 items

length and vc

List Tabs Length and vc				
UNI	Name	VC	LENGTH	Overall Average Grade
	(no filter)	(no filter)	(no filter)	(no filter)
	auto[1].length[0]	auto[1]	length[0]	100%
	auto[1].length[1]	auto[1]	length[1]	100%
	auto[1].length[2]	auto[1]	length[2]	100%
	auto[1].length[3]	auto[1]	length[3]	100%
	auto[1].length[4]	auto[1]	length[4]	100%
	auto[1].length[5]	auto[1]	length[5]	100%
	auto[1].length[6]	auto[1]	length[6]	100%
	auto[1].length[7]	auto[1]	length[7]	100%
	auto[1].length[8]	auto[1]	length[8]	100%
	auto[1].length[9]	auto[1]	length[9]	100%
	auto[1].length[10]	auto[1]	length[10]	100%
	auto[1].length[11]	auto[1]	length[11]	100%
	auto[1].length[12]	auto[1]	length[12]	100%
	auto[1].length[13]	auto[1]	length[13]	100%
	auto[1].length[14]	auto[1]	length[14]	100%
	auto[1].length[15]	auto[1]	length[15]	100%
	auto[2].length[0]	auto[2]	length[0]	100%
	auto[2].length[1]	auto[2]	length[1]	100%
	auto[2].length[2]	auto[2]	length[2]	100%
	auto[2].length[3]	auto[2]	length[3]	100%
	auto[2].length[4]	auto[2]	length[4]	100%

Showing 24 items

length and dest_port

List Tabs				
length and dest_port				
UN1	Name	LENGTH	DEST_PORT	Overall Average Grade
	(no filter)	(no filter)	(no filter)	(no filter)
	dest_port[0].length[0]	length[0]	dest_port[0]	100%
	dest_port[0].length[1]	length[1]	dest_port[0]	100%
	dest_port[0].length[2]	length[2]	dest_port[0]	100%
	dest_port[0].length[3]	length[3]	dest_port[0]	100%
	dest_port[0].length[4]	length[4]	dest_port[0]	100%
	dest_port[0].length[5]	length[5]	dest_port[0]	100%
	dest_port[0].length[6]	length[6]	dest_port[0]	100%
	dest_port[0].length[7]	length[7]	dest_port[0]	100%
	dest_port[0].length[8]	length[8]	dest_port[0]	100%
	dest_port[0].length[9]	length[9]	dest_port[0]	100%
	dest_port[0].length[10]	length[10]	dest_port[0]	100%
	dest_port[0].length[11]	length[11]	dest_port[0]	100%
	dest_port[0].length[12]	length[12]	dest_port[0]	100%
	dest_port[0].length[13]	length[13]	dest_port[0]	100%
	dest_port[0].length[14]	length[14]	dest_port[0]	100%
	dest_port[0].length[15]	length[15]	dest_port[0]	100%
	dest_port[1].length[0]	length[0]	dest_port[1]	100%
	dest_port[1].length[1]	length[1]	dest_port[1]	100%
	dest_port[1].length[2]	length[2]	dest_port[1]	100%
	dest_port[1].length[3]	length[3]	dest_port[1]	100%
	dest_port[1].length[4]	length[4]	dest_port[1]	100%
	dest_port[1].length[5]	length[5]	dest_port[1]	100%
	dest_port[1].length[6]	length[6]	dest_port[1]	100%
	dest_port[1].length[7]	length[7]	dest_port[1]	100%
	dest_port[1].length[8]	length[8]	dest_port[1]	100%
	dest_port[1].length[9]	length[9]	dest_port[1]	100%
	dest_port[1].length[10]	length[10]	dest_port[1]	100%
	dest_port[1].length[11]	length[11]	dest_port[1]	100%
	dest_port[1].length[12]	length[12]	dest_port[1]	100%
	dest_port[1].length[13]	length[13]	dest_port[1]	100%

vc_req

List Tabs			
vc_req			
UN1	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
	auto[1]	100%	58879
	auto[2]	100%	59034
	auto[3]	100%	56212

Showing 3 items

outport_req

List Tabs			
Outport_req			
UN1	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
	outport_req[1]	100%	43314
	outport_req[2]	100%	43578
	outport_req[4]	100%	43704
	outport_req[8]	100%	43529

Showing 4 items

vc_gnt

List Tabs			
vc_gnt			
URI	Name	Overall Average Grade	Score
	(no filter)	(no filter)	(no filter)
auto[1]		100%	41820
auto[2]		100%	41908
auto[3]		100%	90397

Showing 3 items