

Lab 9

HTAX Functional Coverage

Reminder:

Please take a few minutes of your time to evaluate the course anonymously on <https://tamu.aefis.net/> - Your feedback is valuable.

Note: the deadline to submit your course evaluation is **December 4th, 2024**.

Due date

Nov 18, 2024 11:59 PM

Table of content

Academic Integrity.....	2
Introduction.....	2
Design Under Test.....	2
Environment Setup.....	2
To-do.....	4
Deliverables.....	5

Academic Integrity


The following actions are strictly prohibited and violate the honor code. The minimum penalty for plagiarism is a grade of zero and a report to the Aggie honor system office.

- Sharing your solutions with a classmate.
- Uploading assignments to external websites or tutoring websites
- Copying solutions from external websites or tutoring websites
- Copying code from a classmate or unauthorized sources and submitting it as yours

Introduction

In this lab, you will learn to code and measure Coverage for the HyperTransport Advanced X-Bar design specification.

Design Under Test

The DUT is the HyperTransport Advanced X-Bar whose specifications are mentioned in this document:  HyperTransport Advanced X-Bar HTAX Specification.pdf .

Environment Setup

1. Accept the assignment's repository on GitHub Classroom:

<https://classroom.github.com/a/b2SJR55W>

2. Source the setup file.

```
source setupX.bash
cd work
```

3. We have now a full UVM-TB for HTAX design in place (from lab8). Go through all the files in lab9/ directory.

4. TX Functional Coverage: Open tb/htax_tx_monitor_c.sv. You need to complete the covergroups cover_htax_packet and cover_htax_tx_intf.

5. RX Functional Coverage: Open tb/htax_rx_monitor_c.sv. Create covergroup for htax_rx_inf and add at least one coverpoint.

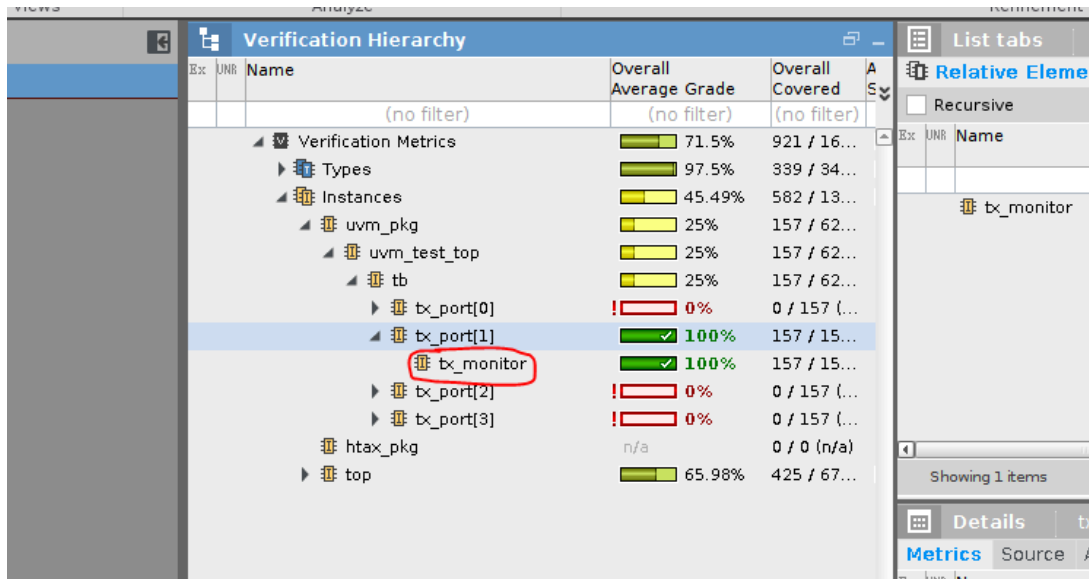
Add as many meaningful cover points as you can think of. There is no limit on number of coverpoints.

6. Simple random test: This test runs sequence - simple_random_seq on port[1] sequencer. Run this test using below command:

```
cd sim
xrun -f run.f +UVM_TESTNAME=simple_random_test
```

7. Coverage in imc: Open imc and load the database. (look for directions in previous lab manuals)

Navigate through hierarchy to find tx_port[1]-tx_monitor coverage.



Ex	UNR	Name	Overall Average Grade	Overall Covered	A
		(no filter)	(no filter)	(no filter)	
▲		Verification Metrics	71.5%	921 / 16...	
▶		Types	97.5%	339 / 34...	
▲		Instances	45.49%	582 / 13...	
		uvm_pkg	25%	157 / 62...	
		uvm_test_top	25%	157 / 62...	
		tb	25%	157 / 62...	
		tx_port[0]	0%	0 / 157 (...)	
		tx_port[1]	100%	157 / 15...	
		tx_monitor	100%	157 / 15...	
		tx_port[2]	0%	0 / 157 (...)	
		tx_port[3]	0%	0 / 157 (...)	
		htax_pkg	n/a	0 / 0 (n/a)	
		top	65.98%	425 / 67...	

Right click on it and select “Cover Group Analysis”. Once you write all the covergroups and coverpoints they will be listed in the tool window.

Overall Covered Grade: 100% Functional Covered Grade: 100% CoverGroup Covered Grade: 100% Assertion C Ed			
Cover Gro.. Assertions			
Cover groups			
Size	Unit	Name	Overall Average Grade
		(no filter)	(no filter)
		cover_htax_packet	100% 147
		cover_htax_tx_intf	100% 10
Showing 2 items			
Items cover_htax_packet			
Size	Unit	Name	Overall Average Grade
		(no filter)	(no filter)
		DEST_PORT	100% 4 / 4
		VC	100% 3 / 3
		LENGTH	100% 16 /
		A*B X_DEST_PORT_VC	100% 12 /
		A*B X_DEST_PORT_LENGTH	100% 64 /
		A*B X_VC_LENGTH	100% 48 /
Bins			
Details cover_htax_packet			
Attributes Source			
Col #	Name	Value	
	(no filter)	(no filter)	
	At Least	1	
	Comment		
	CoverGroup Average Grade	100%	
	CoverGroup Covered	147.0	
	CoverGroup Excluded	0.0	

8. tx_port[1]-tx_monitor coverage: Next you need to modify your tb/sequence(s) (htax_seqs.sv file) to achieve 100% coverage for tx_port[1]-tx_monitor, as shown above. *Hint: You can modify existing simple_random_seq sequence.*

To-do

1. In the **htax_tx_monitor_c.sv**, complete the following tasks
 - 1.1. Coverpoint for htax packet field: vc (include vc=0 in illegal bin)
 - 1.2. Coverpoint for htax packet field: length (Divide range [3:63] into 16 bins)
 - 1.3. DEST_PORT cross VC
 - 1.4. DEST_PORT cross LENGTH
 - 1.5. VC cross LENGTH
 - 1.6. Coverpoint for tx_outport_req: covered all the values 0001,0010,0100,1000
 - 1.7. Coverpoint for tx_vc_req: All the VCs are requested at least once. Ignore what is not allowed, or put it as illegal
 - 1.8. Coverpoint for tx_vc_gnt: All the virtual channels are granted at least once

(The above covergroups are instantiated in constructor and sampled in different part of driver logic.)

2. In the **htax_rx_monitor_c.sv**, complete the following tasks
 - 2.1. Create covergroup for htax_rx_inf and add at least one coverpoint

3. In the IMC, take a screenshot to show you have achieved **100% coverage** for tx_port[1] - tx_monitor (as shown in step 7).

NOTE: You have to add coverage screenshot only for tx monitor coverage and not for rx monitor coverage. Total 2 screenshots (1 screenshot for htax_packet covergroup and 1 screenshot for htax_intf covergroup).

4. Make a lab report that includes your **htax_tx_monitor.c.sv** and **htax_rx_monitor.c.sv** code, UVM summary/simulation output that shows that there are no UVM_FATAL/UVM_ERROR/Assertion failures, coverage screenshot (as described in the previous TO-DO) and name it lab_report.pdf.

Deliverables

Commit and push all your changes to your remote repository.

Your repository must include the following:

- The test directory
- The sim directory containing lab_report.pdf
- The tb directory containing updated files.

Important note: To get full credit, you must upload all the required files and directories and strictly name your files according to the requirements.