Lab 5

HTAX Sequences and Test

Due date

Oct 16, 2024 11:59 PM CST

Table of content

Academic Integrity 2
Introduction 2
Design Under Test 2
Environment Setup 2
To-do 3
Deliverables 4

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 your

Introduction

In this lab, you will learn how to use a base sequence to write complex sequences for different test case scenarios. You will also learn to use sequencers to run your sequences.

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/MSzr01y4
2. Source the setup file.

```
source setupX.bash
```

3. Open tb/htax_seqs.sv. code for Base sequence. Another sequence called Fix Destination Port Sequence is also provided in it. We shall extend all the sequences from htax_base_seq. Take a look at the semantics of "fix_dest_port_seq". This sequence generates all packets on a fixed destination port. A random value is generated using $urandom_range(0,3) and is fixed for all packets.
4. Write the sequences that are mentioned in the TO-DO section. In the tests/ directory, there are three files.
   - test_lib.svh – include all the tests in this file
   - base_test.sv – We shall extend our tests from this base test (Go over the structure once)
   - mix_sequence_test.sv – We will instantiate all the sequences in this test
5. Open – tests/mix_sequence_test.sv. There are three steps to start all sequences (as we did in lab 3) listed as TO DO. Example provided for "fix_dest_port_seq".
6. In this lab, we have a dummy sequencer and a dummy driver code in place. The driver code is written such that it will only print the packet it receives.

7. The initial block in top.sv has the task “run_test()”. This task runs the test declared in the command line argument “+UVM_TESTNAME=<test>” while invoking xrun command. Simulate your design with the command below.

```
xrun -f run.f +UVM_TESTNAME=mix_sequence_test
```

Additionally, you can add “-svseed <seed_number>” as a command line argument to simulate design with a particular seed.

8. Once it is done, go through the xrun.log, which prints all the generated transactions.
9. In the sim/ directory, there is an extra Python script (seq_db.py) that can process this log post-simulation. This script dumps all the packet properties along with the parent sequence name in a CSV format.

```
python3 seq_db.py
```

The output is seq_db.csv. Export it and open it using excel.

```
Sequence, dest port, vc, length, delay
fix_dest_port_seq_, 'h0, 'h1, 'h13, 'h9
fix_dest_port_seq_, 'h0, 'h2, 'h10, 'h4
fix_dest_port_seq_, 'h0, 'h1, 'h3, 'hf
fix_dest_port_seq_, 'h0, 'h2, 'h14, 'h4
fix_dest_port_seq_, 'h0, 'h1, 'h15, 'he
```

The purpose of this script is to help you visualize whether the constraints you set at the sequence level are being honored or not.

To-do

1. You need to write four more sequences (use fix_dest_port_seq as a reference). Below are these sequence names and their intent.

- short_packet_seq – pkt length is between 3 and 10.
- long_packet_short_delay_seq – pkt length is between 40 and 50, and delay is less than 5
- med_packet_fix_vc_seq – pkt length is between 10 and 40, and vc is fixed (either VC-0 or VC-1)
- random_seq – With default constraints

2. Perform the following three steps for all four sequences:

- Create a handle(pointer) for the sequences
- Create UVM instance of sequences using factory

- Start the sequences on the sequencer

3. Raise objection before we start this test and drop objection after all the sequences are run. Wait for 5us after dropping the objection to end the test.
4. Export seq_db.csv using the Python script provided.

Deliverables

Commit and push all your changes to your remote repository.

Your repository must include the following:

- The test directory with the updated changes
- The sim directory with seq_db.csv and the run history
- The tb directory containing the new sequences, with all the appropriate changes.

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