

OVM/UVM Learning Resources

- 1) Download UVM Library, Class Reference and User Guide at [Accelera Web](#)

Also find examples in the library

- 2) OVM/UVM CookBooks – Comprehensive explanations

[Verification Academy CookBooks](#)

- 3) OVM 2.0 Golden Reference Guide

[OVM 2.0 Golden Reference Guide](#)

– Details on class members and methods supported in objects/components

- 4) Verification methodology training links from Verification Academy [here](#)

- 5) Mentor's published Guidelines

[Guidelines on OVM/UVM usage in SOC Verification](#)

- 6) Beginners guide on UVM – [An excellent blog article](#)

- 7) Good papers on creating Stimulus using OVM/UVM Sequences

DVCon2013 Paper – [Sequence on the Wall – Who's the Fairest](#)

DVCon2013 Paper – [Seven Seperate Styles of Using Sequences in UVM](#)

Mentors Guide on – [All about Sequences and Sequence/Driver API](#)

- 8) How to Terminate Tests in UVM?

[UVM Termination Techniques](#)

- 9) Understanding Importance of UVM Factories

[SNUG2012 Paper on Importance on UVM Factories](#)

- 10) Hierarchical configuration of UVM Testbench – [Paper from Synopsys](#)

Course examples for Simple driver/sequencer/agent available for reference

https://github.com/VerificationExcellence/UVMReference/tree/master/course_examples

Exercises/Assignments

1) TLM1 example - Try simulating on edaplayground

Simulate and learn the simple TLM based communication between a producer and consumer component.

- a. Create a simple producer and consumer component and implement the put and get port connection between the producer and consumer
- b. Create an env class that instantiates the producer and consumer and connects between the ports and exports
- c. Create a “module test” as top level which instantiates the env class and calls the run_test method.

Demonstrate using: <http://www.edaplayground.com/x/8Ay>

3) APB Basic Project (Building an APB Testbench)

Step1: In addition to the lecture that explained APB protocol - here is a nice summary of what APB interface protocol is that you will find useful to read and understand first

<http://www.icverification.com/BusProtocols/AmbaAPB.php>

A working copy of complete APB project is available on Git hub for reference
https://github.com/VerificationExcellence/UVMReference/tree/master/apb_project

Also a working copy is available on edaplayground for reference
<http://www.edaplayground.com/x/53i>

However it is advised that students doesn't copy the code as it is before trying out.

Step2: Log onto [www.edaplayground](http://www.edaplayground.com) and use following skelton code to get started

- 1) Go to <http://www.edaplayground.com/x/53i>
- 2) Click on copy on the top to replicate a session for you
- 3) Go through each file and complete the assignment by coding the relevant portions. Refer to the sample reference at location in Step1 in case you need
- 4) Once coded hit "Run" on top menu and resolve any compile issues
- 5) Observe the logs below to see what UVM_INFO messages shows up from driver and monitor

Optional Steps:

1. Create more flavours of sequences and run those sequences to see the behavior
2. Following are few sequences that you can attempt
 - a. Have two random sequences of same 10 sequences started in a fork..join in the top level test
 - b. Create a sequence that does 10 writes and another sequence that does a read of same 10 addresses