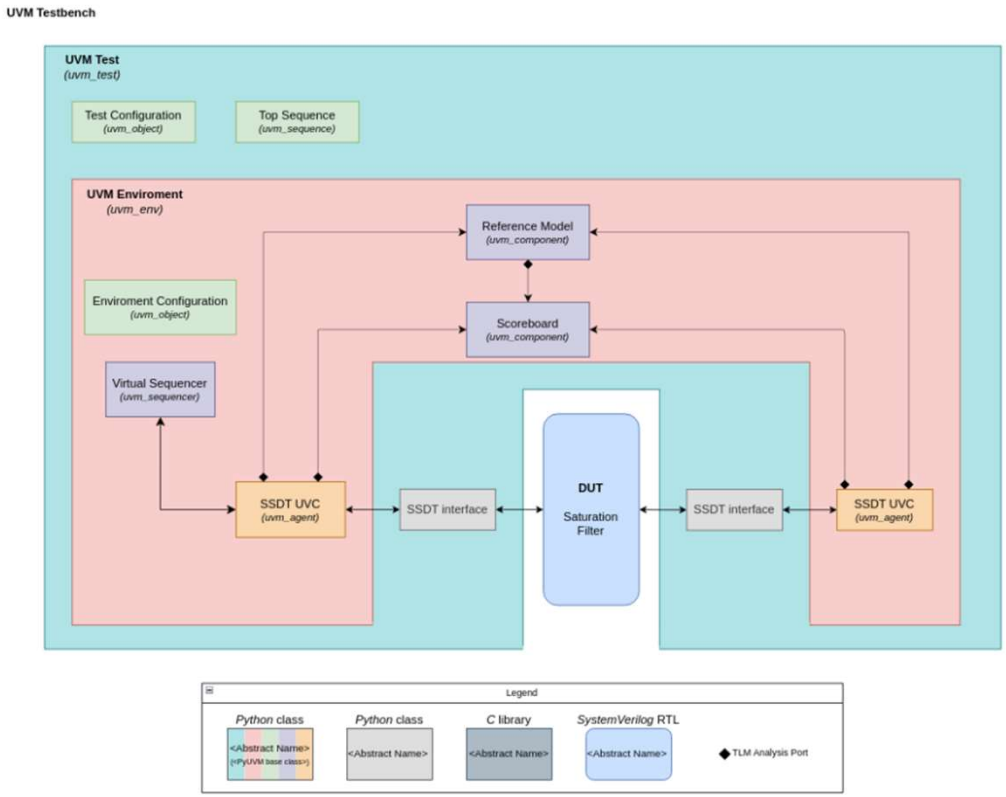
Testbench architecture overview

SyoSil, 2024

Contents

- Testbench architecture diagram
- Base test
- Environment
- Virtual sequence
- Reference model
- Scoreboard

Testbench diagram



Base test

- *uvm_component* which extends from *uvm_test*
- Instantiates and connects:
  - Environment
  - Configuration
  - Virtual sequence
  - Interface classes
- Creation is done in the *build_phase*
  - Use the UVM Factory whenever possible
- Connection is done is *connect_phase*
- Generation of traffic happens during *run_phase*

Environment

- *uvm_component* which extends from the *uvm_env*
- Instantiate the *configuration*, the *virtual sequencer* and the required *uVCs*
- If available, the environment instantiates the *reference model* and the *scoreboard*
- Creation is done in the *build_phase*
  - *Configuration* can be "*get*" from the *configDB*
  - uVCs are also configured in this phase
- Connection is done in the *connect_phase*
  - *sequencers* are connected to the uVCs
  - *TLM* connections are connecting:
    - uVCs monitor to *coverage* classes and/or *scoreboard*
    - *uVCs monitor to the reference model*

Virtual sequence

- extends from *uvm_sequence*
- Can start multiple uVC's *sequences* on the respective *sequencers*
- It runs on the *virtual sequencer*, which handles the uVCs *sequencers*
- Can be used:
  - To generate *data* traffic
  - To generate *register* transactions
  - Many more…

Reference model

- Implements the algorithm which is implemented in the design
- It is connected to the monitor of the uVC which generates the data traffic
- It generates the golden output, used as reference to verify the design

Scoreboard

- The scoreboard is a *passive* component
- It is used to compare the output of the reference model against the output of the DUT, using *queues*
- It reports the mismatches as *errors*