DAY 83 - 100 DAYS VERIFICATION CHALLENGE

Topic: UVM Phases

DAY 83 CHALLENGE:

1. Which are the three main types of UVM phases & the explain each phase they have in detail with syntax:
   i. Build Phases
      - Build
      - Connect
      - end_of_elaboration
   ii. Run-time Phases
      - start_of_simulation
      - run

      Parallel Run-Time Phases

      - pre_reset, reset, post_reset
      - pre_configure, configure, post_configure
      - pre_main, main, post_main
      - pre_shut_down, shut_down, post_shut_down

   iii. Clean-up phases (extract, check, report, final)
2. Which UVM phases refers to the function and which phases are a task?
3. How can you start a UVM phase?
4. Can we have a user-defined phase in UVM?

DAY: 33

Topic : UVM PHASES

Sol"① which are the three main types of uvm phases and the explain each phase they have in detail with syntax :

1. Build phases:

- Build
- Connect
- end_of_elaboration

② Run-time phases :

- Start_of_simulation
- RUN

③ Cleanup phases :

- extract
- check
- report
- final

- pre_reset
- reset
- post_reset
- pre_configure
- Configure
- post_configure
- main
- post_main
- pre_shutdown
- shut_down
- post_shut_down

• uvm_Component class also defines a set of phases that are used to manage the initialization and shutdown of the Component. The following are the different phase of a uvm Component, along with a explanation of each :-

① Build_phase : its used to create testbench component.

• build_phase is the first phase of Component and its used to Create and initialize all of sub-Components of the Component.

• This includes creating any necessary interfaces, sequencers, drivers or monitors & Configuring them with the appropriate settings.

② Connect_phase :

• Connect_phase is used to establish Connection Between the Sub-Components of the Component.

• This includes Connecting interfaces to drivers, sequencers to monitor and so on. The Connect_phase called after the build_phase.

③ end_of_elaboration_phase :

- The end_of_elaboration phase is called after all of the components in the testbench have been created and connected. It is used to perform any final setup or configuration of the component.

Syntax :-

```
function void end_of_elaboration_phase (uvm_phase phase);
endfunction
```

: Component using end_of_elaboration_phase : test.

④ start_of_simulation_phase :

- The start_of_simulation_phase is called at the beging of the simulation, after all of the components have been initialized and configured.
- Its used to perform any additional setup that is required before the simulation begins. Such as setting up design hierarchies, setting initial values and preparing for the actual simulation.

: Component using start_of_simulation_phase : test.

⑤ run_phase :

- The run_phase is the main phase of the component and is used to perform the actual testing.
- its responsible for generating and driving transactions, monitoring the DUT and collecting coverage and other data.
- Run_phase means (start Component functionality) and Component using run_phase : test, driver, monitor, sbd.

⑥ extract_phase : extract_phase use Collect the outputs.

- The extract_phase is used to extract data from the simulation for analysis and reporting.
- This includes collecting coverage data, generating waveform & dumping other data that is needed for debugging & analysis

: Component using extract_phase is test.

⑦ check_phase : its used to Compare outputs.

- The check_phase is used to perform any checks or assertions on the data collected during the simulation.
- This includes checking for errors in the DUT, verifying the correctness of the TB and reporting any failures or errors.

: Component using check_phase is est.

⑧ Report_phase : it is used to report status.

- The report_phase is the final phase of the component & is used to generate reports and summarize the result of the simulation.
- This includes generating coverage reports, performance metrics & other data that is needed for analysis & verification.

Syntax:

```
function void report_phase (uvm_phase phase);
  ....
endfunction
```

⑨ Run_phase : as a scheduled phases:

- Whenever run_phase is called, a set of predefined phase run concurrently.
- These phases are executed during actual simulation runtime when the verification environment functionality is actively testing the DUT.

Sol ② which uvm_phase refers to the function and which phase are a task ?

- only run_phase is a tasks that run concurrently (means time consuming phase) & other phase are function (non-Blocking).
  - ⊙ Build_phase → function
    - i.) connect_phase
    - ii) end_of_elaboration
    - iii) start_of_simulation_phase.
  - ⊙ Run_phase → task
  - ⊙ clean_phase → function
    - i.) check_phase
    - ii) report_phase
    - iii) final_phase

Soln ③ How can you start a uvm phase?

- run_test() method in top module using then starts the functionality of all the components by run_phase of every component automatically.
- instantiate uvm_root as top, allocates memory, then calls top.run_test().

Soln ④ Can we have a user-defined phase in uvm?

- user defined phase in uvm can be inserted within the Run phase and allows us to create and use our own defined phase.
- This can be done with the use of callbacks like exec_task & exec_function as all the phases can be extended either from uvm_topdown_phase, uvm_bottomup_phase & uvm_task_phase:

: Steps for to create a user defined phase:

① Defined a new user defined phase.

```
class user_phase extends uvm_task_phase / uvm_top_down_phase
                                          uvm_bottomup_phase;
endclass
```

② use get_uvm_schedule function to add the new user phase in uvm schedule.

③ use of exec_task or exec_function depending on whether the phase will consume simulation time or not.

④ finally insert the user phase within the prefered runtime_phase.