UVM - Day 21

-: Transaction - Level Modeling (TLM) :-

TLM is used for communication among modules. TLM is the concept in which transaction based methods are implemented, these methods can be used for communication b/w the modules.

UVM TLM ⇒

The UVM provides TLM library with transaction-level interfaces, ports, exports, imp ports, and analysis ports. all these TLM elements are required to send a transaction, receive transaction, and transport from one component to another. where each one plays its unique role.

- TLM interfaces consists of methods for sending and receiving the transaction.
- All different types of TLM ports are used like PIPES to connect between the components.

The UVM TLM library provides,

- TLM1 - The TLM1 ports provide blocking and non-blocking put by value transaction-level interfaces.
- TLM2 - The TLM2 sockets provide blocking and non-blocking transaction level interfaces with well-defined completion semantics.
- Sequence Port - A push or pull port, with well-defined completion semantics.
- Analysis - The analysis interface is used to perform non-blocking broadcasts of transactions of connected components.

TLM1 :-

UVM TLM provides unidirectional and bidirectional,
- TLM interfaces
- ports
- exports
- imp ports
- analysis ports
- FIFO's

- Each TLM interface is either blocking, non blocking or a combination of these two.

Blocking :-

Blocking TLM methods call will not return until the transaction has been succesfully sent or retrieved.

Non - blocking :-

Non blocking TLM methods call attempts to convey a transaction without consuming simulation time.

Combination :-

A combination interface contains both the blocking and nonblocking variants.

UVM TLM Methods :-

The TLM interface class declares all the methods required to perform communication.

- put :- Put method is used to send a transaction to another component.
- get :- get method is used to retrieve transaction from another component.
- peek :- peek method obtain a transaction without consuming it
- try_put :- try_put method is used to send a transaction to another component without blocking the execution.

- **can_put:—** can_put() method call returns 1 if the component is ready to accept the transaction, otherwise, it returns 0.
- **try_get:—** try_get() method is used to retrieve transaction from another component without blocking the execution.
- **can_get:—** can_get() method call returns 1 if the component can get transaction immediately, otherwise, it returns 0.
- **transport:—** Calling <port>.transport (req, resp) method executes the given request and returns the response in the given output argument.
- **nb_transport:—**
  Calling <port>.nb_transport (req, resp) method executes the given request and returns the response in the given output argument.
- **Analysis :—**

  write – Calling <port>.write (trans) method will broadcasts a transaction to any number of listeners. It is nb.