UVM

1. Uvm important concept like factory,resource db and config db majorly use static properties and methods for their implementation.
2. Without this concept of static properties and static methods. It would not possible with factory.

Parameterized class:

* Provides reusability of classes
* SV supports both type and value parameterization.
* UVM Implements lot of base classes as parameterized classes, so that they can reused for various type and value parameterization.
* Parameterized classes are very important for UVM to work efficiently

Parameterized macros

Macros with arguments are widely implemented in UVM

`uvm\_component\_utils: Parameterized macro

Widely used across UVM TB coding

1. Object & component registration to the factory
2. Reporting
3. User defined TLM Classes

SV-UVM Mapping

1. PROGRAM -> ROOT
2. Sub\_env -> agent
3. Display -> uvm\_info/warning/error
4. Generator -> sequencer
5. Bfm -> driver
6. Test is a task -> test is a component
7. Tx/pkt -> item
8. Scenario/stimulus -> sequence
9. Mailbox -> tlm1.0/tlm2.0(port,imp,export)
10. New -> create/new
11. $finish-> user need to call

OBJECTION:

1. Simulation is completed when all objections raised are dropped
2. Phase.raise\_objection(this)
3. Phase.drop\_objection(this)
4. Phase is coming from uvm\_root,hence uvm\_root keeps track of all objections raised and dropped.

SET DRAIN TIME:

Phase.phase\_done.set\_drain\_time(100,1);

Even after all objections are dropped,we wait for 100 ns for simulation to finish.

Ex: if 3 components of TB raised objection,all those objection are dropped at 1200ns,if drain time is 100 ns,simulation will be finish at 130ns.

Drain time is set in the test run\_phase

Objects don’t have phase,since only components can have phase

How do we raise/drop objection?

Objects don’t have build\_phase/connect phase etc

Solution: starting phase

Phase in which sequence will be started

How sequence will get the staring phase value?

It is decided by test

Objection gives smooth finish to the simulation

OBJECTION SCENARIOS

1. No objection are raised and dropped in TB?

Simulation ends at time=0

1. Some of the objection are never dropped in simulation?

Simulation will hang.it will by default end at 9200sec unless user has passed UVM\_TIMEOUT argument.

1. $finish is called before all objections are dropped?

Simulation will finish since user is manually asking to end simulation

* 1. Why objections are never called in driver,monitor,scoreboard etc

Driver has forever loop implemented.it will never complete,any objection raised,will never dropped hence it will result in simulation hanging.

Where we can raise and drop objection?

Objection are raised in components which does not have forever functionality.

Ex: objection cant be raised in driver,monitor,coverage,SBD etc

Objection can be raised in test,sequence

Objections raised and dropped in test run\_phase

Objections raised and dropped in sequence body task.

What if a raised objection is not dropped?

Simulation will hang and times out at 9200sec,unless user has given UVM\_TIMEOUT

Raise objection in sequence.

Ex:

Class ahb\_base\_seq extends uvm\_sequence(ahb\_tx);

`uvm\_object\_utils(ahb\_base\_seq)

Task pre\_body();

If(starting\_phase!=null) begin

Starting\_phase.raise\_objection(this);

End

endtask

Task post\_body();

If(starting\_phase!=null) begin

Starting\_phase.drop\_objection(this)

End

endtask

endclass

Arguments are two types

Those which apply for whole testbench

Arguments is all upper case letters

Those which apply only for specific hierarchies of the testbench

Arguments is all lower case letters

Reporting methods:

Uvm\_report\_object implements various methods for message reporting

Uvm\_report\_info

Uvm\_report\_warning

Uvm\_report\_error

Uvm\_report\_fatal

Factory

Factory is an instance of uvm\_factory class.

It is singleton class

Factory is a global scope variable