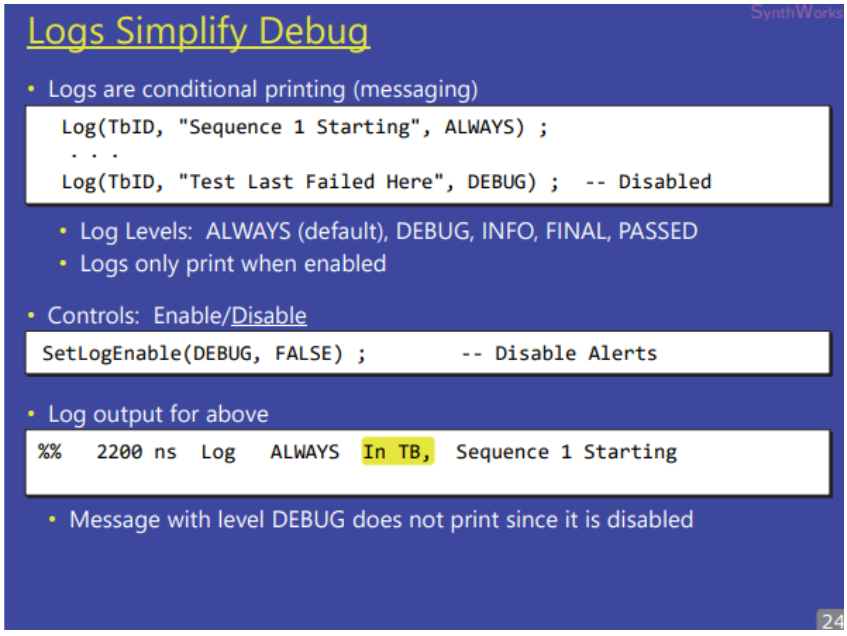
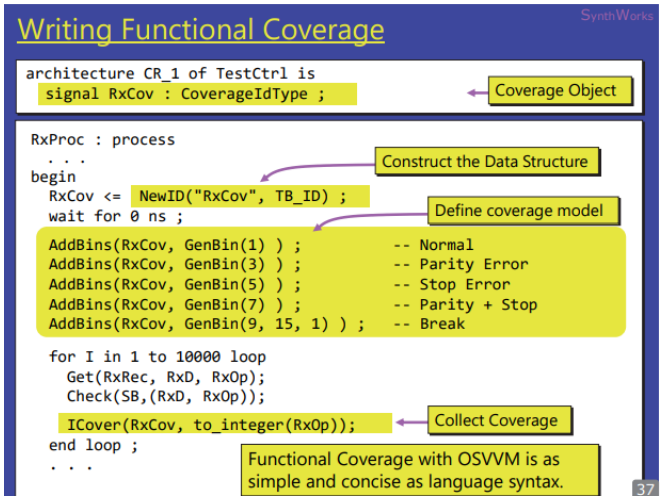
SUMMARY of OSVVM

- The OSVVM verification framework has a structure similar to SystemVerilog, including verification components, a test sequencer, and separate test case architectures.
- The implementation of the OSVVM verification framework brings together the DUT, verification components, and test sequencer using structural code similar to RTL.
- In summary, OSVVM implementation combines DUT, verification components, and test sequencer with RTL-like structural code.
- **In OSVVM, the transaction API is implemented as VHDL procedures that handshake with the verification component using the transaction record and retrieve the results.**
- OSVVM's easy approach defines common transaction interfaces and APIs for interfaces that perform similar transactions using model independent transactions (MIT).
- The benefits of OSVVM MIT include accelerating the development of verification components and test cases, supporting co-simulation, simplifying documentation, and providing more information in user guides.
- OSVVM test scenario, send, get and check transactions and affirmations (checks) use for test, affirmations show pass/fail and count errors.
- OSVVM alerts, protocol checks easy do, error messages give, errors count, and different levels (FAILURE, ERROR, WARNING) alerts create.
-

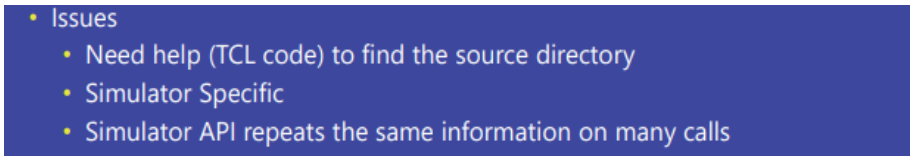


- OSVVM test finalization, test finish check, timeout check, reports create, affirmations check, and test pass confirm.
- FIFO's directed and constrained random test compare, and realistic stimulus create and similar items wide variety ideal benefits emphasize.
- It is difficult to synchronize the randomization logic of these two processes.

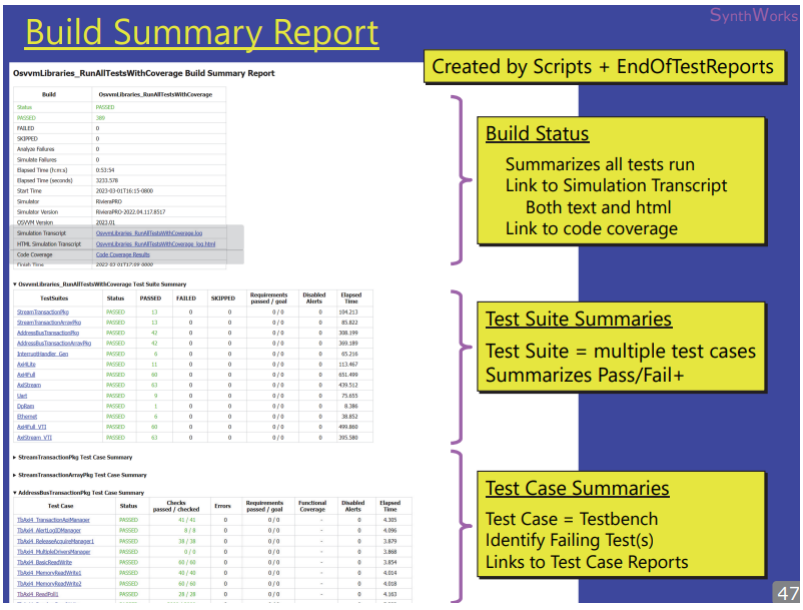
- OSVVM scoreboards use, sender (TxProc) change even, receiver (RxProc) process same keep, control easy do.
- OSVVM functional coverage, test plan track, random tests what do show, different coverage types with relations track, and code coverage missing parts complete.
-

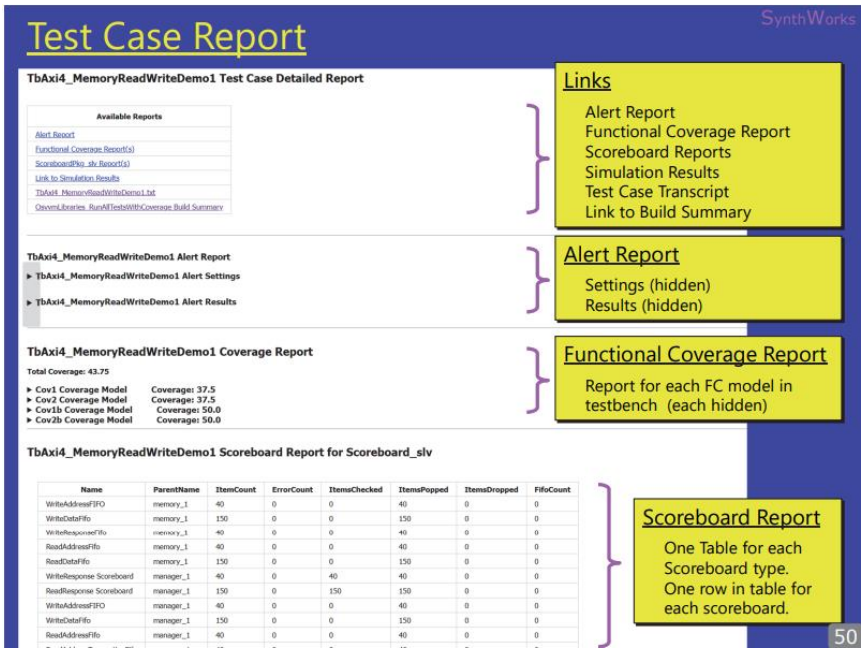


- Before OSVVM scripts;



- OSVVM unmatched test reports create, test completion message, summary reports, requirements summaries, test case reports, HTML'ized simulation transcript, JUnit report and more give.
- Build finish when one line mini report create, errors have first see place this.

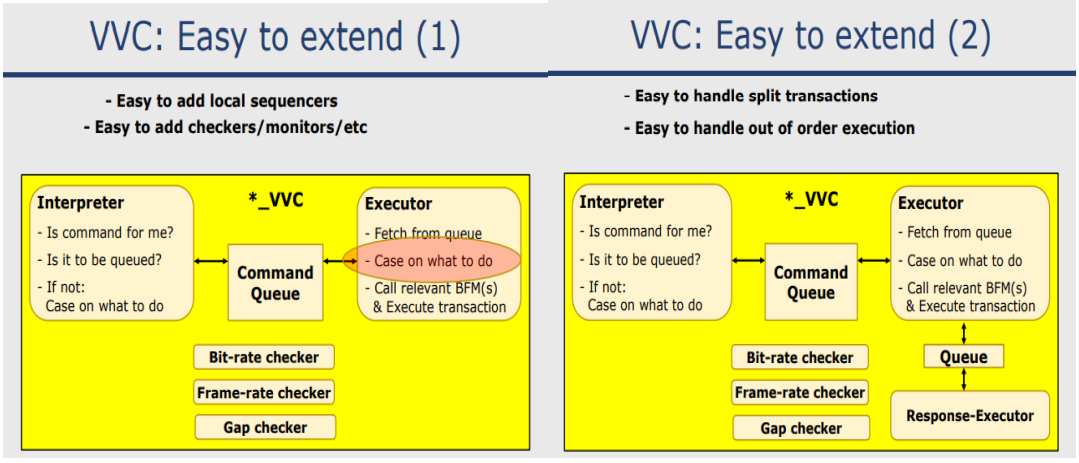




- OSVVM HTML'ized simulation transcript, It makes debugging easier by allowing you to review things quickly.
- In many ways, OSVVM offers a more sensible approach to testing.

SUMMARY of UVVM

- UVVM is an open-source VHDL verification methodology.
- UVVM, simple data communication verify for BFMs and utility library use UART and SBI over data send, control and wait show, error situations emphasize, and wide interface support give.
- In UVVM, VVCs make interface verification easy, test scenarios modular and reusable do.
- **In UVVM, BFM to VVC transition, test sequencer direct DUT instead VVC with communication make, test scenarios more modular and expandable do, delay insert, command queuing, completion detection like extra features give.**
-



- **UVVM VVCs, multiple interfaces simultaneously manage, reusable and expandable, single sequencer from controlled can, cycle corner cases target, split transactions and out of order protocols easy handle, broadcast and multicast use, better overview and maintenance give.**
- UVVM generic scoreboard, VVC based test environment expected and actual data compare, statistics keep, queue manage, generic data types support, configuration records and counting features have.
- UVVM enhanced randomization, readability increase, maintenance easy make, and better coding invest big gain give.
- UVVM functional coverage, bins use by coverage target reach make and transition coverage also support.