This article describes how incorporating LLI Questa verification intellectual property (QVIP) can yield a host of benefits, including faster, more flexible verification and easier debugging. In this article, we will discuss how the sequence items are useful to generate the stimulus based on multiple layers of LLI QVIP, how the combination of LLI QVIP based on coverage-driven methodology and protocol-capturing XML plans can boost verification completeness, how the combination of protocol assertions and error injection method is useful to generate the error scenario and check the behavior of the LLI design, and how the scoreboard is useful to check the data integrity between LLI design and LLI QVIP.

OVERVIEW

Questa Verification IP (QVIP) enables reuse of the components and build environment. Other features include layered verification methodology supported by highly configurable and feature-rich transactors, a protocol assertions monitor, advanced debug support and comprehensive functional coverage. LLI-based design IPs require a complex verification system in which standalone Verilog testing can kick-start verification planning but is not sufficient for complete verification. What’s needed is a well-planned and executed QVIP and methodology that address the following questions: Is the captured protocol in design IP verified? Which error scenarios are verified? Have you covered all required scenarios? Can you provide a progress report to your manager? These challenges are not new for verification engineers. However, complex verification projects often force teams to do more planning. If they don’t, their verification engineer can easily get lost in technical detail, which slips the project schedule, jeopardizes the quality and increases the risk of re-spin.

The combined use of LLI QVIP, Universal Verification Methodology (UVM) and coverage-driven verification metrics for all compliance items can provide much needed predictability.

LLI QUESTA VERIFICATION IP

LLI Questa Verification IP (QVIP) is a UVM-based multilevel abstraction verification IP that provides fast and flexible verification along with easy debugging of LLI designs. It is compliant with all the released versions of LLI provided by Mobile Industry Processor Interface (MIPI) Alliance. LLI QVIP provides a Reference M-PHY MODULE Interface (RMMI) SystemVerilog interface to hook up with the M-PHY at the physical layer, which transmits the serial data towards the remote LLI via remote M-PHY present in the system. LLI QVIP supports the configurable RMMI with the following:

• width: 10, 20 or 40-bit interface
• maximum number of lanes in any direction

The above parameters are static so it’s required to pass these parameters at the very start of the day. The number of lanes can be up-configured and down-configured as per requirement during the test but can’t exceed the maximum lane count parameter.

The figure below shows the typical LLI subsystem environment, where LLI QVIP is connected to the local M-PHY, which is connected with the remote M-PHY through the serial interface. Now, the LLI design is connected with the remote M-PHY at RMMI and LLI QVIP is connected with the local M-PHY at RMMI.
LLI QVIP implements the transaction layer, data link layer and PHY adaptor layer and has its own configuration space that can be configured to emulate the design behavior. In terms of configuration variables, it can also emulate behavior of the sideband signals, debug sideband signals and IPC signal. It supports the features like back-door access of configuration registers, coverage implementation of an LLI checklist to measure checklist coverage, and a scoreboard for end-to-end data checking. Various useful sequences are provided for easy/quick generation of stimulus at each layer, and high configurability at each layer allows for altering the QVIP behavior to verify the design for different scenarios and perform error injection for all relevant errors at different layers.

COMPONENTS OF LLI QVIP
The following are the main components of the QVIP.

a) Sequence Items
LLI QVIP provide various sequence items corresponding to each layer. Sequence items consist of all the members corresponding to each transaction, methods and constraints to generate the stimulus. Sequence items of LLI QVIP generate stimulus in packet, frame and phit format based on TL, DL and PA layer. LLI QVIP provides a top-level sequence item that can be used to send all types of transactions supported by LLI to different layers. This sequence item is the best utility to initiate any transaction irrespective of the layer. Among the sequence items:

1) Top Level Sequence Item:
   • lli_component_txn: used to send all form of transactions supported by all layers of LLI.
2) Transaction Layer Sequence Item:
   • lli_component_packet: used to send transactions supported by LLI at Transaction layer
3) Data Link Layer Sequence Item:
   • lli_component_frame: used to send frames supported by LLI at Data Link layer
4) PHY Adaptor Layer Sequence Item:
   • lli_component_phit: used to send phit supported by LLI at PHY Adaptor layer

The figure below shows the members of the sequence item.
Note: there are many other sequence items corresponding to functionality defined in the LLI specification, including some for error injection scenarios.

**b) Sequences**

LLI QVIP provides a rich set of sequences that are very useful to generate any kind of scenario. Most of the sequences are used to generate the LLI transaction. These sequences can be easily used to generate the complex and corner case scenarios for the LLI design verification. Examples include:

1) lli_write_req_seq: used to send Best Effort Write request or Low Latency Write Request depending upon the channel id given by the user with random address, ord_id, length of transaction & opcode_type
2) lli_read_req_seq: used to send Best Effort Read Request or Low Latency Read Request depending upon the channel id with randomized address, ord_id and length.
3) lli_random_req_seq: used to generate the random number of transactions.
4) lli_mounting_seq: used for mounting the Master or Slave with respect to the variable m_master_not_slave.

**c) Functional Coverage**

Functional coverage is the most important aspect of any verification project. Coverage helps to understand what portion of the features have been covered and what are the remaining features or corner cases. Based on that information, it should be very easy to dig out the remaining testcase or to enhance the existing test suite to cover what's missing. LLI QVIP provides an XML coverage plan to cover the entire feature list, supported by the LLI Protocol that can be linked to Unified Coverage Database (UCDB) dumped by the Questa simulator. The UCDB contains information like code coverage, cover directives, cover points and assertion coverage that can be merged with an XML plan with all coverage results in the form of UCDB, which is accessible both via log file and GUI.

LLI QVIP comes with a predefined coverage collector corresponding to each layer of the LLI. There are coverpoints and crosses defined for each individual layer in the separate coverage collector. It helps to understand whether or not all coverpoints and crosses related to a particular layer exist in the plan. LLI QVIP also provides configuration to enable/disable the particular covergroup and coverpoints belonging to a particular covergroup. It's very easy to add or remove the coverage for any required layer, as shown below.

```plaintext
<config_handle>.enable_tl_cvg[LLI_MASTER] = 1'b0 ; // to disable particular covergroup
<config_handle>.enable_tl_cp[LLI_MASTER]
[LLI_PKT_TYPE] = 1'b0; // to disable particular coverpoint of a covergroup
```

The figure below shows achievement of 100% coverage for the covergroup of the LLI transaction layer in the Questa Analysis window.
LLI QVIP provides a coverage test suite to cover the entire feature list specified in the XML test plan captured from the LLI specification. The suite divides tests for multiple layers. All layers have random and directed transaction tests to cover all features with respect to the LLI specification. To cover the transaction and data link layer scenarios, LLI QVIP provides an example named “tl_dl” that can be run from the LLI QVIP agent by integrating the LLI design environment. This will help to cover all scenarios and features specified in the XML test plan for TL and DL layer.

d) Scoreboard
A scoreboard is one of the critical verification components that checks data integrity for memory-based transactions. A scoreboard is a TLM component and care should be taken not to activate it on a cycle-by-cycle basis but rather at the transaction level. In LLI QVIP, the scoreboard contains four analysis ports, two for transmission of LL and BE traffic from LLI QVIP and two for reception of LL and BE traffic by the LLI QVIP. It compares the data received during the Read Response transaction with data previously written by the Write Request transaction from LL and BE traffic class. If there was a previous write to the address, and there is a mismatch between the data read and the data expected, then an error is issued.

e) Protocol Checking
The protocol checker is also one of most required features present in Verification IP. It helps in debugging errors due to illegal activity on the bus or to any bad transaction. Whenever illegal activity occurs, the protocol checker flashes an error message. It is also helpful in checking the design behavior during an error scenario. LLI QVIP provides a mechanism to enable/disable all assertions or any particular assertion using the API. (See code below.) LLI QVIP provides many assertions for monitoring illegal activity during LLI design verification. These assertions are divided for multiple layers, which helps the user identify which layer of LLI design created the erroneous condition.

```c
<bfm handle>.set_config_enable_all Assertions(0); // to disable all assertions
<bfm handle>.set_config_enable Assertion_index1(LLI_PA_RCVD_WRONG_SEQ_NUMBER, ’b0); // to disable particular assertion
```

The figure below shows the assertions hit during the test case run in the Questa Assertion window highlighted in red.

LI QVIP also provides two mechanisms to inject these errors (either by sequence items or by config variables) to check the erroneous behavior of the LLI design under these conditions. These config errors variables are available by layer to provide granularity.

CONCLUSION
This article describes how LLI QVIP can be useful in LLI design verification by using sequence items, sequences, coverage collection, assertion checking, scoreboarding and more. These techniques can be used in your verification process to save time and improve the quality of results. Mentor Graphics QVIP spans multiple engines, simulation, formal and acceleration. LLI QVIP is a highly configurable verification IP, which can be used in any SV UVM environment.
VERIFICATION ACADEMY
The Most Comprehensive Resource for Verification Training

20 Video Courses Available Covering
• Intelligent Testbench Automation
• Metrics in SoC Verification
• Verification Planning
• Basic and Advanced UVM
• Assertion-Based Verification
• FPGA Verification
• Testbench Acceleration
• Power Aware Verification
• Analog Mixed-Signal Verification

UVM and Coverage Online Methodology Cookbooks
Discussion Forum with more than 4400 topics
UVM Connect and UVM Express Kits

www. verificationacademy.com