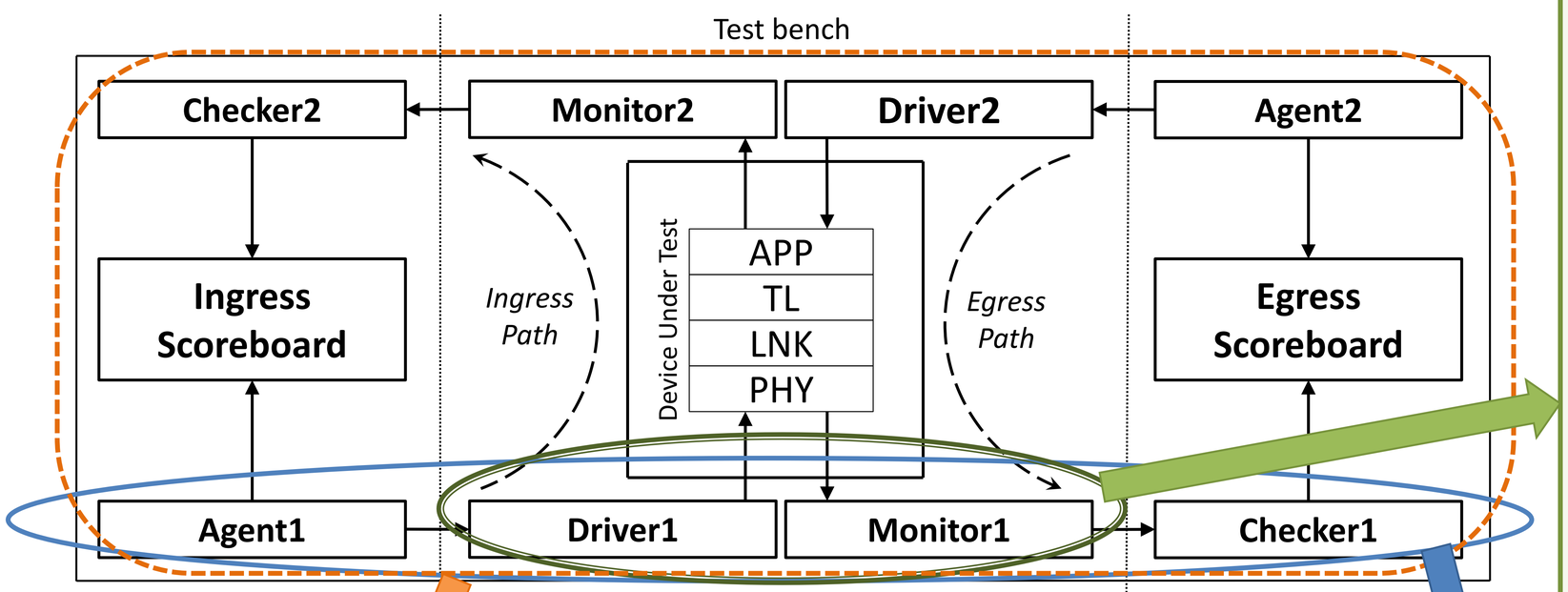
VIP Shielding

Jeremy Ridgeway, Karishma Dhruv
LSI Corporation, Inc.
San Jose, CA

Abstract Third party verification intellectual property (VIP) is often leveraged to achieve testing and coverage goals more quickly. However, also in the interest of time, the verification architecture is not sufficiently shielded from a particular VIP vendor. VIP shielding delineates hard boundaries between the in-house verification environment and the third party VIP. While the acceptance of Universal Verification Methodology (UVM) provides a springboard to VIP shielding, simply relying on the common methodology is not enough. In this paper we show how we incorporated a third party PCI-Express (PCIe) VIP in our verification environment. We sufficiently shielded the environment to enable swap-out should management deem it necessary.

Where does VIP fit in my simulation?



Option 1: Let them do it all

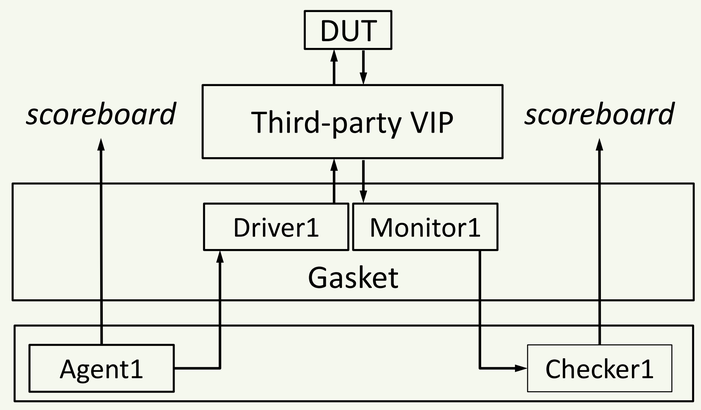
- VIP Provider:
  1. Builds components for DUT proprietary interface,
  2. Re-uses VIP provider components for standard interface,
  3. Owns the test bench and all testing.
- Change of VIP provider requires a complete new test bench.

Option 2: Connect at the standard interface

- VIP connects through standard interface.
- Test bench and testing is performed in-house.
- Translation between in-house and VIP-specific data classes for scoreboards:
  1. VIP-specific typed scoreboard, or
  2. In-house typed scoreboard.
- Change of VIP provider requires rework of test bench to match new VIP.

Option 3: VIP Shielding

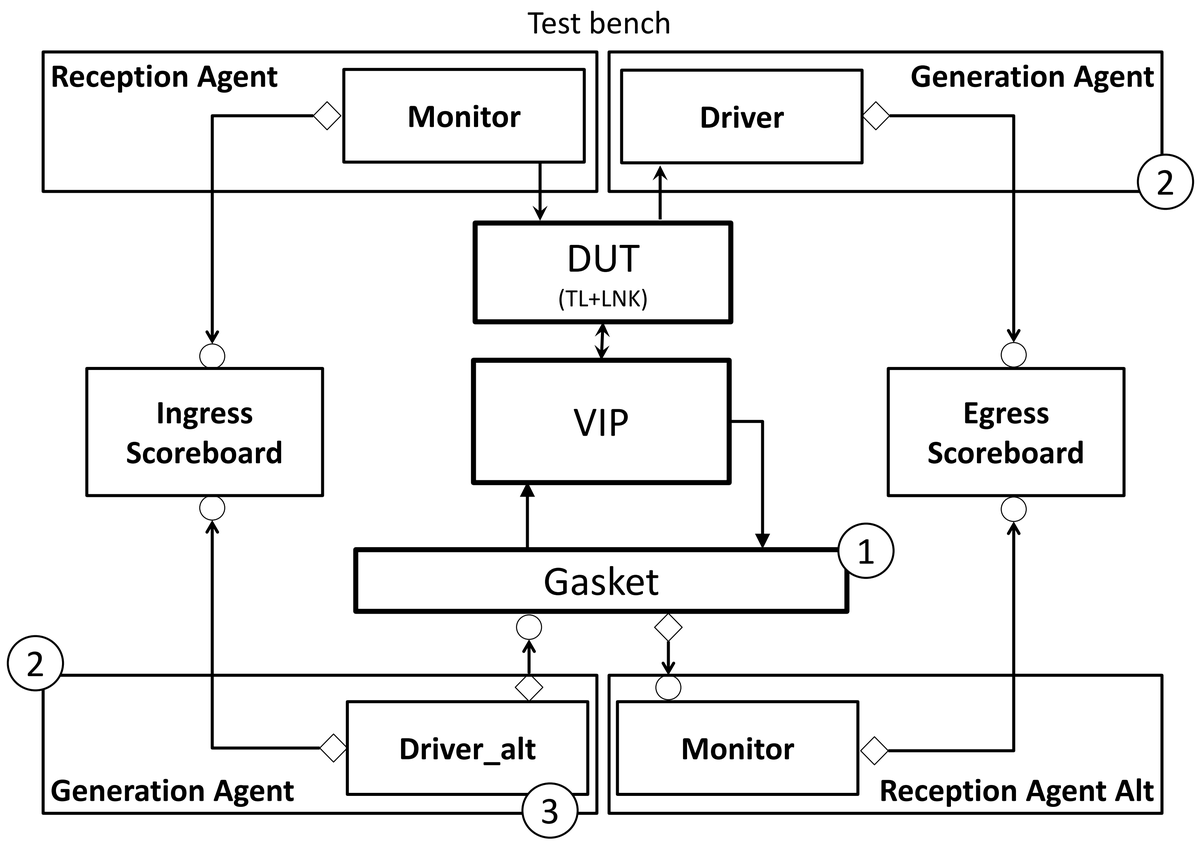
- VIP connected through standard interface.
- Gasket shields VIP from test bench:
  1. Translates in-house and VIP-specific classes,
  2. Provides an interface to VIP to test bench.
- Test bench and testing performed in-house.
- Change of VIP provider requires re-connection to gasket.



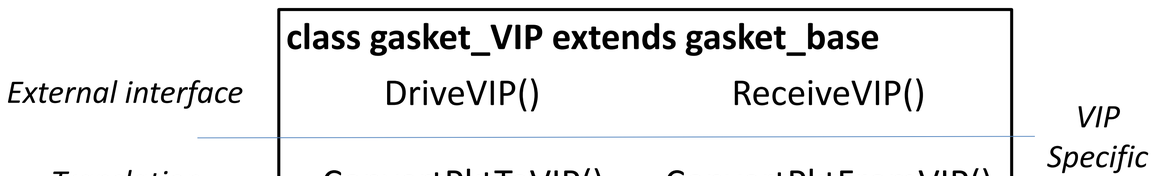
Legend

APP – Application Layer
TL – Transaction Layer
LNK – Link Layer
PHY – Physical Interface
◇ – UVM analysis port
◯ – UVM analysis implementation port

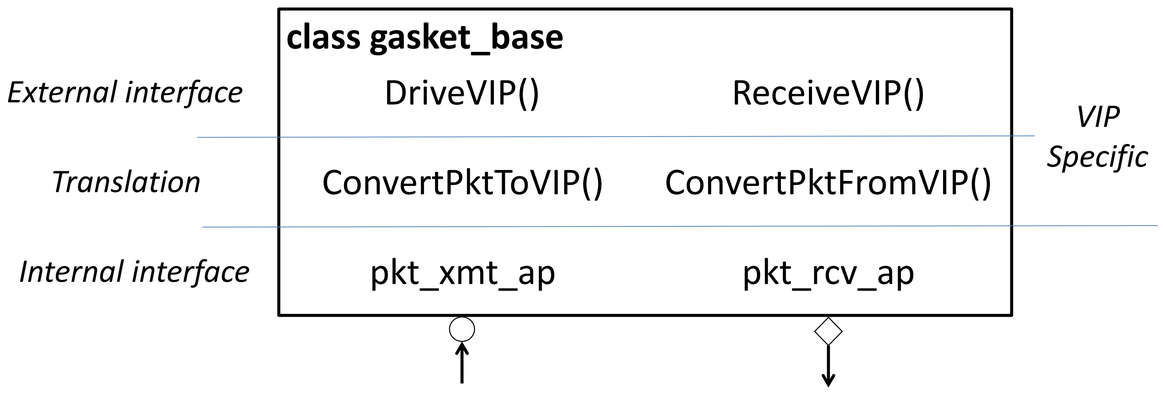
Case Study: PCI-Express Verification



1. Shielded test bench from VIP instance with gasket class.
2. Re-used generation agent for egress and ingress paths.
3. Override driver in generation agent through UVM factory to connect to gasket.
   - Conceptually, reception agent can be re-used, too.



The VIP-specific gasket implements the pure virtual functions *only* to connect the VIP to the test bench. The DriveVIP/ReceiveVIP functions are a driver and monitor.



The base gasket fully defines the internal interface functions, while the VIP-specific portion is left pure virtual. The test bench at large accesses the VIP *only* through gasket interfaces.

Conclusions

We presented an approach to VIP inclusion in verification architecture that allows for vendor or major revision change through VIP shielding. We found that a gasket is sufficient to shield the verification environment from the specificities of the VIP. Furthermore, implementation of the gasket is relatively straightforward utilizing features of UVM and SystemVerilog.

However, a trade-off in effort and responsibility does exist. With partial data path testing (VIP connected to one side of the DUT) and partial VIP stack (using some but not all the verification layers), the onus is on the in-house environment for all verification activities. The VIP can assist in protocol checking, but as its scenario layer is disabled it cannot produce random stimulus. However, if the VIP can be utilized in both extremities of the DUT protocol layer, then the verification onus returns to the VIP. Considering the unknown nature of the VIP market and management decisions, having a plan for VIP change without disrupting verification (too much) is a wise idea.