When beginning a new design it’s common to evaluate how to build a verification infrastructure in the quickest amount of time. Of course it’s never just quick to deploy, verification also has to be complete enough to improve confidence in the design. Rapid bring-up and improving the quality of your design are excellent goals. However, you should not forget that your environment should be efficient to use during the verification process. This is where you will spend most of your time, slugging it out day after day. Arguably, debugging design bugs is one of the most time consuming tasks of any project. Transaction Level Modeling (TLM) will change the way you think about debug productivity, especially if you have recently experienced the long and difficult task of deciphering PCIe’s training sequences, data transfers and completion codes at the pin level.

Who has resources to learn new verification technology rapidly enough to have an impact on the verification quality of an upcoming design? Most teams begin by looking at schedules, ongoing support tasks and self-determine that they don’t have enough time or resources. However, everyone who has standard interfaces like PCIe can answer “I can improve my verification productivity with Mentor Verification IP”.

Mentor VIP enables productivity in deployment, adherence to the protocol for quality and improved time to debug design errors. Mentor VIP is a self-contained high level verification environment based on OVM or UVM with a built-in protocol based test plan that is fueled with high level test scenarios that deliver functional coverage based test plan tracking. Mentor VIP delivers on the promise of UVM/OVM high level verification infrastructure for validating protocols like PCIe.

There is a growing community of SystemVerilog and UVM/OVM knowledgeable engineers, but who has the significant resources needed to create an advanced verification environment? Such an environment is already available to validate standard protocols like PCIe. Simply creating a test plan for a complex protocol like PCIe alone is too large for many teams; you also have to generate test scenarios. This is just part of what goes into architecting a productive verification environment.

The answer to having enough time to improve your productivity is to re-use the verification infrastructure that meets your requirements that is already pre-built for you.

If you have experience debugging PCIe design in either pre or post silicon you have experienced the very daunting task of debugging at the pin level. Tales of deciphering long training sequences, data transfers and completion codes are common when people share their difficult PCIe debug experiences.

One of the main benefits of high level verification is Transaction Level Modeling (TLM). TLM is not just used to implement high level test scenario generation. You should require that your verification environments also support debugging with TLM viewing if you want to maximize your productivity.

Mentor VIP uniquely extends to debugging of high level PCIe transactions within Questa’s simulation environment. A common debug scenario is speed negotiation during LTSSM training. Suppose you are designing PCIe Gen2 device and at initial training the Link reaches L0, at speed change to 5GT/s it fails and the link drops back to 2.5GT/s, a significant loss in performance. This is a classic PCIe design bring-up error. Why did the negotiation to 5GT/s fail? Do both sides of the link exit idle and send the proper training sequences at 5GT/s? Is the upstream device not sending TS2 ordered sets or is it the downstream device? Did you set the proper speed in the PCIe data rate field? What is the value of the N_FTS Field and are you being held back by this? Or perhaps it’s a software bug, did you validate the data rate in the packet is the proper speed rating? Figure 1 is a screenshot of a PCIe Gen2 device in LTSSM link training using the pre-built PCIe Mentor VIP and Questa waveform viewer. From this view you can easily see the N_FTS field has the low number of 0Ah and your data rate field value of 86h is set for Gen2. Armed with this information you can easily make bug hunting decisions. How long would this take you to verify deciphering the serial bus?
How do you gain this level of debug productivity?
It's relatively straight forward.

There are two possibilities. In the first scenario you already have an existing PCIe environment. In this case you can simply attach a PCIe Mentor Verification IP in monitor mode. Mentor VIP will just monitor the PCIe interface and reconstruct the activity at the TL, DL and PL layers. You get the transaction activity in the wave window to debug with a complete history of all activity. This data is also fed to a coverage collector which gives TL, DL and PL level coverage.

In the second scenario you use Mentor VIP either as a RootComplex or EndPoint. In this case the built in monitor and protocol analyzer gives you the transaction details and coverage collection.

Figure 2 shows the SystemVerilog/UVM source code needed to add a PCIe Mentor VIP monitor to your existing environment to monitor your PCIe design.

PCIe is a complex environment. In the previous step we have connected the Mentor VIP to the environment. What remains is to configure it to match parameters to the DUT. Examples are LTSSM parameters, expected speeds of operation, abstraction level of packets the designer wants to observe etc. Figure 3 (on the following page) shows a snapshot of the configuration.
Next you connect the monitor to the design under test (DUT) as shown in Figure 3. Debugging efficiency is only one aspect of a more productive Verification environment. You can easily configure the PCIe Mentor VIP to an active role so you can generate tests with the same efficiency illustrated in the debug scenario above. With these relatively small editions your PCIe environment is connected to a complete monitor, protocol checker and coverage collector.

In the case where Mentor VIP is either the RC or EP, Figure 2 will be modified to ensure Mentor VIP drives the correct end based on whether it is the RC and EP. In Figure 3 the agents will no longer be passive but appropriate sequences will be run based on the desired behavior. You automatically get the monitor, coverage collection and protocol checker.

In conclusion, while thinking about your next project’s test plan, and begin to architect your verification environment you should seriously consider all the options available to help you improve your Verification Productivity with Mentor VIP as illustrated with this PCIe example.
Figure 4: Functional Coverage results and Test Plan Tracking