Beyond the Compliance Checklist:

How End-to-End Verification IP helps you retake control, improve predictability and ensure higher quality SOCs and IP

By Pete Heller and Erez Kovshi

While the trend to use more and more design IP has considerably reduced design effort per gate, it has had the exact inverse effect on functional verification effort. In fact, since integrating multiple design IP blocks is now the norm, verification has become the dominant task and source of risk in SOC projects. To cope with this challenge and contain these risks design and verification teams recognize they need help.

The majority of design and verification teams today use verification IP (VIP) in one form or another. In fact, VIP is essential for any complex protocol or bus standard (e.g., PCI Express or AMBA AHB/AXI). Teams achieving the greatest success combined VIP use with a verification reuse strategy and a methodology that begins with an initial plan and goes all the way through to full verification closure.

Further contributing to verification IP's ascendance is the need to ensure compliance with the complex protocol's checklists. This is a major undertaking in itself and also must span the full verification process.

VIP's end goal is to improve the verification process while reducing your verification project's risk profile. However, the decision as to which VIP to use has only gotten more complicated. Just as automobiles range from stripped down to high end models, several VIP classes now exist. Since VIP is not created equal it is critical to select not just any VIP, but the *right* VIP that balances your needs and resources against risks. When the wrong VIP is selected *teams commonly face* project delays and even outright project cancellations.

This article will help you optimize VIP selection decisions. It will familiarize you with the various classes of VIP and the key issues to consider in your VIP selection.

Key Considerations When Making Your VIP Selection

Since different classes of VIP vary significantly in functionality it is essential to know what capabilities you need. Start the selection process by identifying the critical requirements for the verification project. Listed below are the most common questions that to consider.

- ❑ Will this verification effort be a "quick-and-dirty" integration where you trust the blocks or will you perform a complete functionality verification?
- ❑ How will you reuse VIP when moving from block to chip or to system level verification?
- ❑ How will you ensure compliance for any complex protocols?
- ❑ Will you simultaneously verify multiple protocols?
- ❑ How will you specify your verification goals and measure verification progress?

Depending on how you answer these questions narrows the range of applicable VIP. For example, "quick-and-dirty" integration requires far less capable VIP than does a complete verification. Simple integration can be done with a Bus Functional Model (BFM). Somewhat more complex projects can succeed with "Testbench VIP". On the other hand, IP development and SOC projects place far greater stress on the VIP. They need End-to-End VIP that provides a plan and clear metrics for measuring completeness

and a compliance solution and that spans the full verification process from architectural modeling to block design and on to full chip/system verification. It also must spans the full set of verification engines including formal analysis, simulation, acceleration and emulation.

It is also important to consider your future needs. Even if you don't need many of these capabilities now, it often makes sense to select VIP with headroom to retain your investment as your needs grow.

Matching Your Needs Relative to Verification IP Capabilities

There are three primary classes of VIP. They are BFMs, "Testbench VIP", and "End-to-End VIP". Each class of VIP has its own features and tradeoffs. They are characterized in Figure 1 below.

Figure 1. Characterizing VIP Classes

	BFM	Testbench VIP	End-to-End VIP
Spans full verification process and all verification engines			✓
Fully calibrates, measures and reports on compliance completeness			✓
Ensures reusability from Block to Chip To System Level			✓
Provides plan, metrics and coverage to automatically manage and report on closure status			✓
Automatically generates context-sensitive stimuli			✓
Generates basic stimuli		✓	✓
Drives signals on bus (BFM)	✓	✓	✓

BFMs and Testbench VIP will serve the need for basic block connectivity checkout. For anything more complex End-to-End VIP is needed. While a variety of suppliers offer commercial VIP, you'll find that most semiconductor and systems companies internally develop only BFMs.

Assessing the Value of End-to-End VIP

Automatic Generation: Abstract information to a human usable level

Even a relatively straightforward protocol like AMBA AHB presents a huge complexity challenge to the verification team. A simple bus read has hundreds of permutations that must be captured both in the human readable verification plan document and in the actual verification process itself. To comprehensively test such a protocol manually would require a massive, labor intensive verification plan and process. The limitations of human ability to capture all the verification scenarios make the process even riskier. This is one example of how End-to-End VIP, or a universal verification IP or component, provides a superior result to internally built solutions. These high end universal VIPs abstract the protocol details to a level that humans can effectively manage. Instead of being forced to track millions of protocol permutations (e.g., "did the bus read from location FFFCA while the grant was enabled during a buffer overflow condition") they

provide a higher abstraction level. With this type of VIP engineers will instead ask “have all reads been exercised?”

Some readers may be thinking "most of the permutations don't matter since they’re essentially the same as others already verified". While it’s true that performing a read from address A is not far different than reading from address B, but it is still important in the big picture. Human testers will always miss combinations and edge values thereby weakening verification completeness.

The only approach that enables complete functional coverage employs **context sensitive automatic stimulus generation**. This ensures that all the important configurations and permutations have been stimulated, even infrequent device conditions such as error conditions. For example, automatically generate stimuli for the full range of potential behaviors including error injection.

Furthermore, while simulation cycles are becoming less expensive, they aren’t free (and never will be). Therefore it’s important not to waste them. End-to-End VIP directs the verification engines to avoid protocol functionality not in the specific design as well as already verified functionality. This is a major advantage of using a functional coverage approach. It addresses the entire process from plan to verification closure.

Retake Control: Using an Automatic Plan To Closure Process

Imagine you’re blindfolded and dropped into an open field miles from any city. How would you find your way home? You don’t know where you are and you don’t know where you’re going. That is the conundrum many verification teams find themselves in. They’re charging ahead yet they don’t have a clear map of the territory. Adding to their woes, such teams are unable to concisely and/or accurately convey to their management where they are in the verification process or when they'll finish.

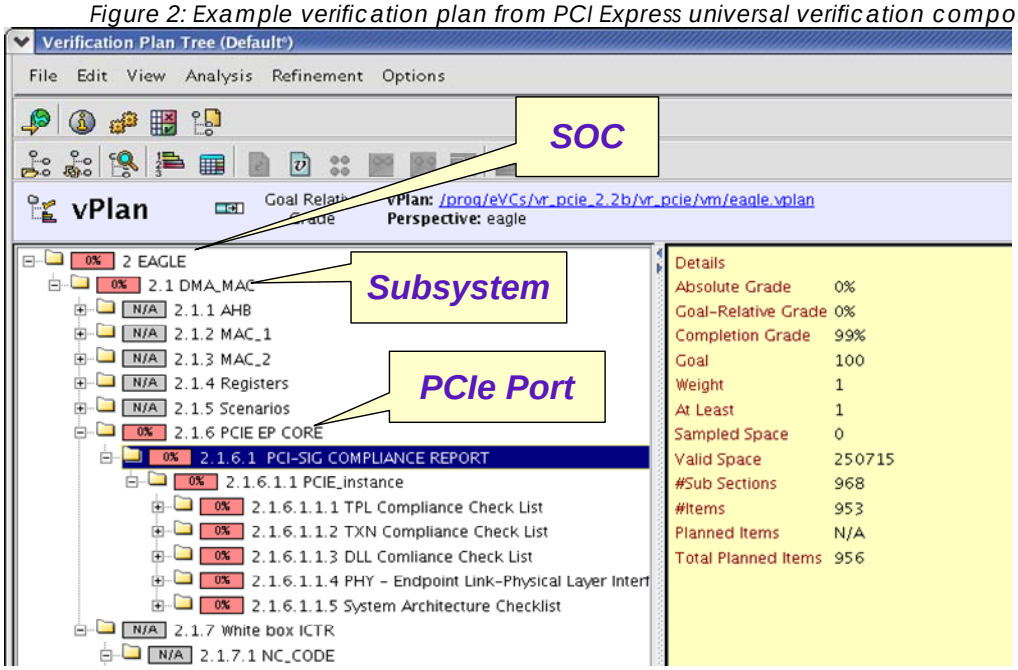
To address these issues End-to-End VIP must deliver three key elements.

1. End-to-End VIP must provides a clear **definition of closure** including metrics that everyone on the team understands. This requires defining a DUT functionality matrix spelling out what is and what is not to be tested. The matrix is then codified into an executable verification plan.
2. It must provide a way to **observe and measure** the verification results relative to the metrics. This enables everyone to understand where the verification stands relative to closure.
3. A **reporting mechanism** is needed so that the team and their management know precisely where the verification process stands. They can also accurately predict how much more time and resource reaching closure will require.

While a plan to closure process is necessary it is still not sufficient. The verification process must be **automated** to successfully verify multi-million gate SOCs. Even if you had the thousands of man-months needed, the complexity of managing thousands of individual test cases is beyond the capacity of manual testing.

Automation capabilities planned into VIP address three essential elements. First, they provide an executable verification plan (vPlan). It is both a human readable and machine readable document that spells out the functionality matrix to be verified (see Figure 2 below). Second, they add in automatic stimulus generation to ensure that each component of the functionality matrix is exercised. And third, they provide the coverage points and coverage metrics to enable you to assess and report on verification completeness.

Figure 2: Example verification plan from PCI Express universal verification component



Verification teams also commonly struggle with the competing tasks of finding bugs and reaching coverage closure. End-to-End VIP alleviates this struggle by automating both. For example, they provide failure triage to find bugs. And, when not finding bugs, they work to deliver maximum coverage. Therefore End-to-End VIP enables each additional simulation seat to find more bugs and/or to increase coverage.

Maximize efficiency: Employ a Reuse methodology so VIP can be used at each verification stage

Verification IP is all about reuse. VIP must be usable (and reusable) by expending only minor effort. This is true of first usage or when moving from block to chip to system level verification. To achieve this goal requires a significant investment in a reusable VIP architecture and a reuse methodology. This can be supplied either by the design team or by the VIP supplier but it must be in place to reap the full benefits of VIP reuse.

This is not just theory. Users employing a market proven reuse methodology together with End-to-End VIP have reported 50-100X productivity gains when creating and reusing their verification environments.

Go Beyond the Checklist: Fully Calibrate, Measure and Report on Compliance

Achieving protocol compliance is often an important part of SOC verification. To help you reach compliance many protocols have associated compliance checklists. While completing such checklists is valuable, checking all the boxes does not guarantee that the design is truly protocol compliant, much less ensuring that the device is fully verified. For example, to truly achieve PCI Express compliance requires that you go beyond the checklist. It requires that (a) an implementation of the checks be provided; (b) closure for each check is calibrated up front (i.e., completeness metrics must be provided for each); and (c) an automatic reporting mechanism must be in place.

These are essential aspects of End-to-End VIP and are necessary to answer the key compliance questions:

- What compliance items have **not** been verified?
- Have all scenarios described by a particular compliance item been covered?
- Will the standard test cases verify the functional customizations specific to the application,
- Can you provide a progress report to your manager and other teams?

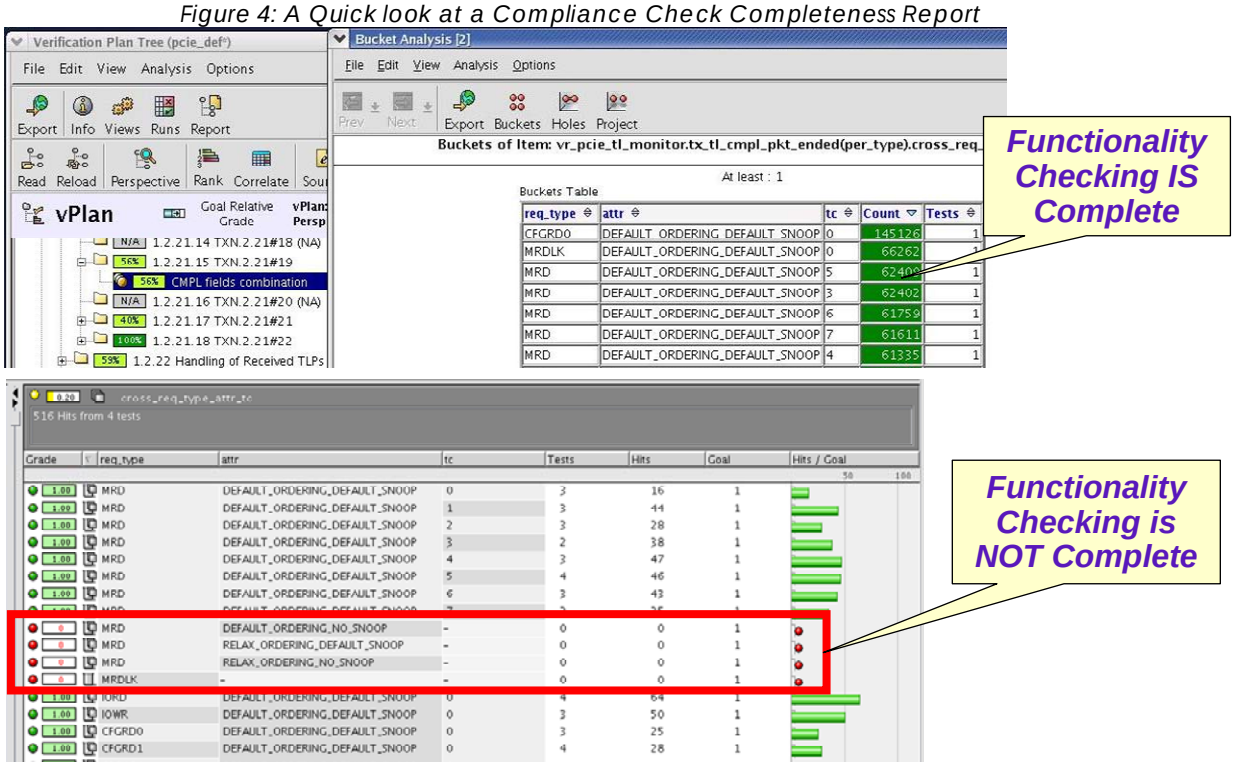
Let's examine a PCI Express Transaction Layer check to exemplify the shortcomings of checklists. **TXN.2.21#19** states that "Completions headers must supply the same values for the Requester ID, Tag, Attribute and Traffic Class as were supplied in the header of the corresponding request". Figure 3 below identifies what the checklist does not provide and what you'd need to supply.

Figure 3: Checklist Shortcomings Example

What the checklist does NOT provide/do	**What you'll need to do**	**Example Logic Required**
Doesn't provide completion handling logic that the check depends on	Create an executable implementation of each check	"Check all completions fields with the corresponding requests"
Doesn't define which completions need to be check compliant	Define the completeness criteria (e.g., there are 52 valid combinations out of 256 possibilities for TXN.2.21#19)	"Check must be done for ALL the request kinds"
Doesn't report which completions were and were not checked	Implement status measurement and reporting	

End-to-End VIP supplies the implementation, the completeness criteria, the coverage, and the reporting mechanisms for you throughout all phases of the verification process. This saves you time and, since it is pre-validated, lets you avoid the problems associated with first-time-used software. For example, a universal verification component provides assertions to check the validity of the data and a functional coverage mechanism to track all the values that have and have not been generated by the DUT. Pairing this infrastructure with the supplied verification plan provides up-to-the-minute reporting on the completeness of any/all compliance checks and the overall verification goals. See the completeness reporting example in Figure 4 below.

Figure 4: A Quick look at a Compliance Check Completeness Report



Conclusion

The need for planning and reuse in SOC projects has exploded along with their skyrocketing complexity. This has made the new standard of End-to-End VIP a critical component of project success.

To achieve your overall verification goals and manage the risks inherent in complex SOC and IP development projects End-to-End VIP, also known as universal verification components, that meet the following requirements must be employed:

- Automatically generate context-sensitive stimulus
- Provide a plan and metrics to automatically manage and report on closure status
- Ensure reusability from Block to Chip To System Level
- Fully calibrate, measure and report on compliance completeness, and
- Fully span the verification process and all verification engines.

With End-to-End VIP, a complete block to system level verification process, and a proven verification methodology, design/verification teams can achieve their quality, predictability, timeliness, and efficiency goals.