

## Closed Loop Requirement Verification

by Anupam Bakshi, Agnisis Technology

### Abstract

The paper talks about a new unifying methodology that ensures closure to Verification planning and execution. Using this methodology and enabling tools, design/verification engineers and managers have a complete handle on the verification process. In a typical flow, a verification plan is created in Excel, Word, XML etc. and imported into the Simulation tool, as the verification process gets underway, the synchronization to the original plan is at best done manually. Further, you need help with understanding if the original requirements are being verified at all. This new methodology/tool, takes away the guessing game. In one *document* you describe the Verification Plan, check the latest Verification Status, know when you are done and what you are done with. This methodology works with Mentor’s UCDB and advanced verification technologies.

### The Problem:

With regard to verification, we have heard the question before “Are we done yet?” Design/Verification Engineers need to know, Managers want to know and executives often ask where all the Verification effort is going and when the work will be completed. There are several tools/methodologies in the industry that can tell us with certainty how much of the code has been covered, how much of the functionality has been exercised.

However, shouldn’t the question really be, “Are we done with verifying the original requirements that we had established?” In other words, how close are we to achieving the goals we had setup initially to build the system according to the specification?

As the chip complexity grows, the need to make sure teams and resources are working towards the original requirements becomes critical. Requirement traceability is not just something that must be achieved in order to get the DO-254 certification; it is becoming a must-have methodology in order to get to market fast while using optimal resources. It shouldn’t be considered a burden, but a means to focus the team on the real objectives and get a clear understanding of where we are compared to our design verification goals.

Often times, the managers and executives on the team need to know what the status of the verification work is. Where are we in terms of the Verification plan that was put together months ago? Are the requirements being verified, are we getting any closer to achieving our objectives, or is it a case of diminishing returns. This need for transparency in the verification process is crucial for managing large scale verification projects.

More often than not teams report on their local achievements and the big-picture is lost in myriad of statistics, fancy reports and meetings.

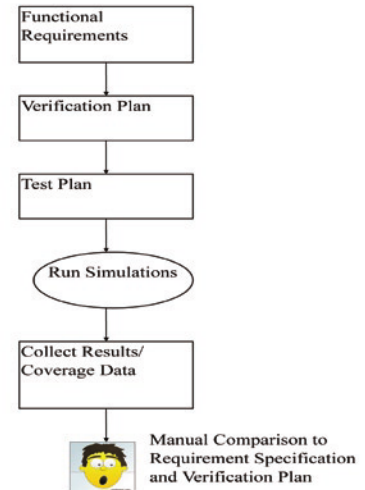


Figure 1: Manual Verification Flow

### Our Solution

In our work as verification consultants, we needed a central place where all requirements, the verification plan and all associated verification progress was kept in an easy-to-access synchronized state, causing the least burden on the actual verification process, the verification environment and on the engineers. We needed a methodology that is not seen by the design and verification teams to be a chore, rather something that efficiently drives the process forward and helps in achieving our design verification objectives.

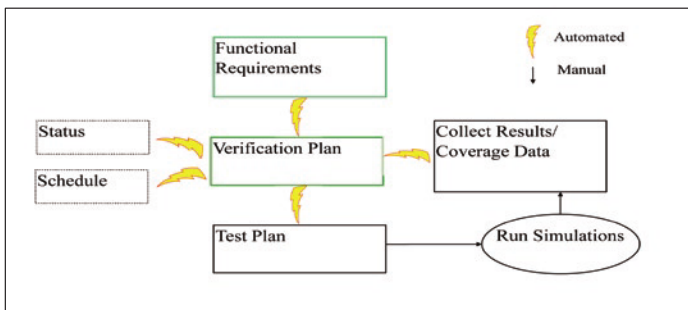
We have come up with a new methodology that revolves around a central *active* design document that was put together by the Design Team. This *active* document is used to capture all the requirements that the design team and the architects set out to implement. The Verification team refers to this *active* document quoting specific requirements and creates a Verification Plan. The beauty of this process is that as the Design Document evolves so does the Verification Plan. The two constantly remain in sync.

### The Verification Plan contains the following:

- pointersto the design document detailing the requirements

- the actual coverage points as per the requirements
- details of various presumed or implemented assertions
- test benches
- details of test

As the requirements get modified, the Verification Plan automatically gets the updates. Verification engineers can quickly note the changes made to the plan and evaluate effectiveness of the plan and make suitable changes. Since the latest up-to-date requirements are right there in the same document, it gives a context and an anchor point for all design and verification activities.



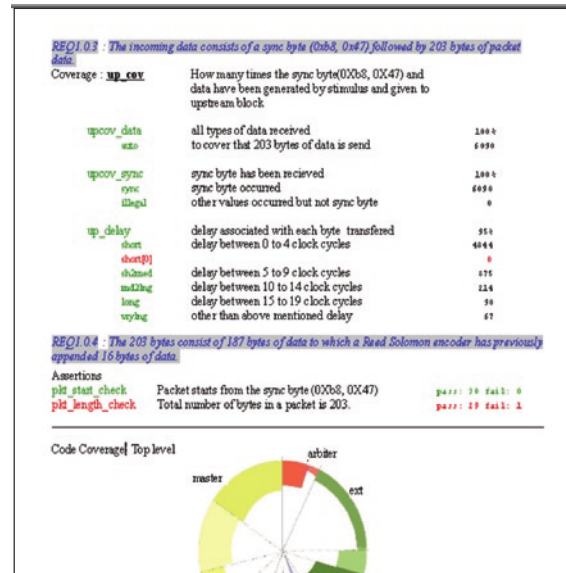
**Figure 2: Closed Loop Requirement Verification Methodology**

We further instrumented this *active* document such that the results from the verification runs were tallied within the same document itself. The verification plan is then the focal point for the design and verification teams, as well as the management and executives. Everyone is able to relate to this document. Designers see their requirements that they have designed to, verification engineers can see their progress in verification space; and management/executives can see the overall progress, all in the same document. This is highly beneficial as just a look at the document clearly reveals where the holes in the verification plan are. Further, a closer examination also shows how to spend the least cycles in Verification for getting the desired features verified.

The active document as shown in figure 3, has requirements imported from the design document, the verification plan has been created in the same document. Further code coverage, functional coverage and assertion results are annotated onto the same document, in a clear and concise way.

Summary results clearly indicate what the status of the requirement verification effort is. It shows the useful information without cluttering it with gobs and gobs of numbers. This gives a clear idea about where the verification holes are and how much work still needs to be done. This leads to more informed schedules and resource allocation.

In principle, this methodology would work with any Vendor's tools. However, Questa's Verification Management facilities, especially the UCDB and its associated xml2ucdb, vcover and coverage utilities,



**Figure 3: Snippets of an active Verification Plan.**

gave us all the hooks we needed to first send test data to the simulation environment and then get access to all the coverage data.

This methodology has many advantages, once all parties have agreed to use it. It was a central rallying point for the entire team, so any team member could see the effects of changes made to either the design or the verification plan. The team could collectively decide where more coverage or assertions were needed, and new members of the team found it advantageous to get up to speed quickly about their role in the verification process. Management and executives were able to open one single document and see the status without any ambiguity or delay.

Having the requirements in close proximity to the Verification plan helps us understand the relationship between the two. This automated flow, eliminates the need to do a manual comparison of the often changing design requirements to the verification plan. In addition the close proximity of the verification results to the requirements and the plan, give a warm and fuzzy about how well the original requirements have been verified with the current effort. Any short-comings can be promptly addressed.

This methodology brings the requirement out of the closet to the forefront of design verification, and helps close the loop on requirement verification.

*About the Author: Anupam Bakshi is the Managing Director at Agnisys Technology Pvt. Ltd. Agnisys provides expert design/verification services and develops EDA tools. Anupam has spent close to two decades creating automation tools for Design/Verification process and managing large scale verification projects. He can be reached at ab@agnisys.us.*