Advancing traceability and consistency in Verification and Validation*

Walter Tibboel, NXP Semiconductors, The Netherlands (walter.tibboel@nxp.com)

Martin Barnasconi, NXP Semiconductors, The Netherlands (martin.barnasconi@nxp.com)

***Abstract*—This paper introduces a structured Verification and Validation (V&V) flow, supported by easy tooling to keep design requirements, test specifications, and test implementations consistent. It offers a practical traceability flow between the design specification and verification objects. Furthermore, this V&V traceability flow offers a clear way to back-annotate the results into the test specification to measure verification coverage and closure. The flow and tooling have been successfully applied in NXP and demonstrated its value by keeping requirements, test items and tests consistent and traceable throughout the entire V&V process. This resulted in a more effective and efficient V&V process and increased product quality.**

***Keywords*— Verification & Validation; requirements-driven verification; traceability; coverage; test specification**

I. INTRODUCTION

In Verification & Validation one of the main challenges is to gain maximum confidence in reaching the V&V targets as part of the product qualification process. Reaching the V&V targets mean covering all specified requirements and functionality, without overseeing the less obvious causes of errors. A well-known mechanism is to start from the product or design requirements when creating the test specification and test items. These test items should be again consistent (and traceable) with the test implementation.

In practice however, we often observe –during the course of the project development– that inconsistencies between the specification and implementation arise, because different persons (owners), in different environments, at different times need to make adaptations or refinements to the requirements, test items or test implementations. Despite the fact that change control mechanisms and traceability measures are in place, inconsistencies creep in and are not always easy to trace, as comparison between these elements is not straight-forward.

The presented V&V traceability flow makes use of templates in Excel and text files, which are often the most natural choice to capture requirements, test items and test results. As shown in Figure 1, design requirements are owned by the architect and stored in one Excel. The V&V lead has its own Excel which also contains the requirements, but his task is to add the corresponding test specification, where each test item specifies what should be tested. For the test execution, a test engineer can maintain a dedicated Excel defining the test implementation by means of test cases. In the functional verification domain, a test case is a simulation file containing the implementation of a number of test items. The flow consists of multiple (hierarchical) levels, each having a specific owner.

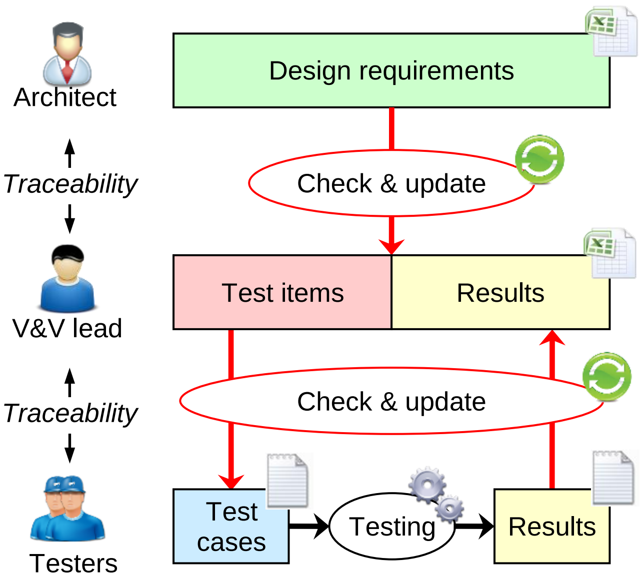

Figure 1 Abstract view of the V&V traceability flow

* This work is partially sponsored by the European Commission in the CATRENE CA703 OpenES project.

The uniqueness of this flow and tooling is to keep these various Excel documents consistent in a controlled way, while maintaining a clear traceability between the various items. This means not all updates are done at once. This, so called “check & update” operation is implemented to check design requirements against test item specifications, or test items against test cases, and only update it when the owner of the spec is confident with the change. It offers clear ways to identify and update differences between the sheets, including incorporating pass/fail results from test execution into the test specification Excel sheet, resulting in a practical dashboard for planning and tracking of the verification process.

Furthermore, the flow supports the capture of both parametric and function verification items and results, but also validation results from real measurements. It has therefore demonstrated its value in analog as well as digital centric design teams, enabling the introduction of a consistent set of verification and validation documents, targets and above all completeness.

The paper is organized as follows. Section II describes the related work in the area of tools and flows available for requirements and test management. Section III will further detail the V&V traceability flow, followed by a proof-of-concept using the proposed flow and tooling in section IV.

II. RELATED WORK

A lot of related work exists on requirements management [1]. In addition, commercial tooling for requirements management is e.g. IBM Rational DOORS [2], which supports traceability between different (hierarchical) modules. Some projects also use DOORS to specify test items, being properly traceable to the requirements. DOORS is not well connected to the engineering environment for test execution. Without such proper connection, the V&V coverage after test execution has no relation to the initial requirements. DOORS can export and import comma-separated (csv) files. This Excel interface can be used by the check & update mechanism proposed in this paper, as depicted in Figure 2.

A solution which has a proper connection to the test execution world is the open source project TestLink [3]. It supports requirements management, test specification, test execution and reporting. TestLink is web based using a SQL database. It is customizable by means of user-defined fields and is accessible via an API. Despite of the customization features, TestLink is aimed for software projects and therefore less attractive for a hardware-oriented V&V traceability flow.

The vManager tool of EDA vendor Cadence [4] supports a regression cockpit for functional verification. The traceability of spec items and the corresponding coverage are coordinated via an executable verification plan (vPlan). This planning part is able to identify changes in the requirement sections in pdf documents. Currently, the focus of vManager is on digital-centric functional verification planning & execution in a semiconductor context and does not address the execution of tests in the mixed-signal or validation domain.

A well-known meta-data format in the semiconductor industry is IP-XACT, also known as IEEE Std. 1685-2009. The IP-XACT standard provides XML schemas to describe IP meta-data and a tool interface to operate on databases supporting IP-XACT meta-data. This enables IP re-use and flow automation. Magillem is a leading EDA vendor in the area of IP-XACT tools [5]. Magillem is applying similar technology for content management and traceability [6], e.g., to manage requirements. Their Meta-X XML schema is intended to describe content meta-data and a tool interface is available to operate on Meta-X databases. Content that can be handled are structured documents in formats such as DOCX, XLSX, DITA, and IP-XACT. Relations between content are captured and maintained in Meta-X. Magillem is currently introducing this technology for requirements management.

Traceability can also be done using Excel-links (cell dereferences) or macro’s. These kind of Excel approaches can be setup quickly, but for the longer term they are not maintenance friendly. The Excel links and formulas are very local and implicit, and get corrupted easily. The use of (Visual basic) macro’s in Excel is restricting to mainly Excel-only, which is too limited for a complete V&V traceability flow. Initially the macro’s are attractive in having full access to all Excel features, but can easily become outdated when moving to a new Excel version. For small projects and limited traceability this approach works fine.

In this paper, the focus is on connecting existing solutions and working practices, creating a mature V&V traceability flow; where needed Excel is used to store meta-data. In general, moving from Excel to a (SQL) database makes the solution more scalable, but potentially less flexible. In this paper, consistency between the levels is maintained by introducing an independent Check & update tool, which uses a stable API for Excel read/write. Next to Excel file support there is generic text file parsing support. Next section will explain this flow in more detail.

III. V&V TRACEABILITY FLOW

The V&V traceability flow covers different levels: Design requirements, test item specification and execution. This chapter has 3 sections: First the overall flow will be explained, next the Check & update mechanism is discussed in more detail, and the last section is about the configurability of this flow, one of the important aspects to gain acceptance from potential users.

A. Flow overview

As explained in the introduction, each level in the flow has a specific owner using dedicated data (attributes) for that level. The traceability has impact only on the shared data attributes, as indicated in Figure 2 by the color scheme. One key attribute is the design requirement ID to link a test item (red) to a design requirement (green). Next to this ID, there are other shared attributes like requirement name, description, minimum spec value, etc.

In a similar way, the test item specification is synchronized with the executable test implementation files (blue). Next to the executable test sources, the regression cockpit has administration of the test items, which requires synchronization with the test item spec. The test results logging (yellow) contain test item identifiers, such that the Check & update mechanism can synchronize a summary of the test results into the test spec.

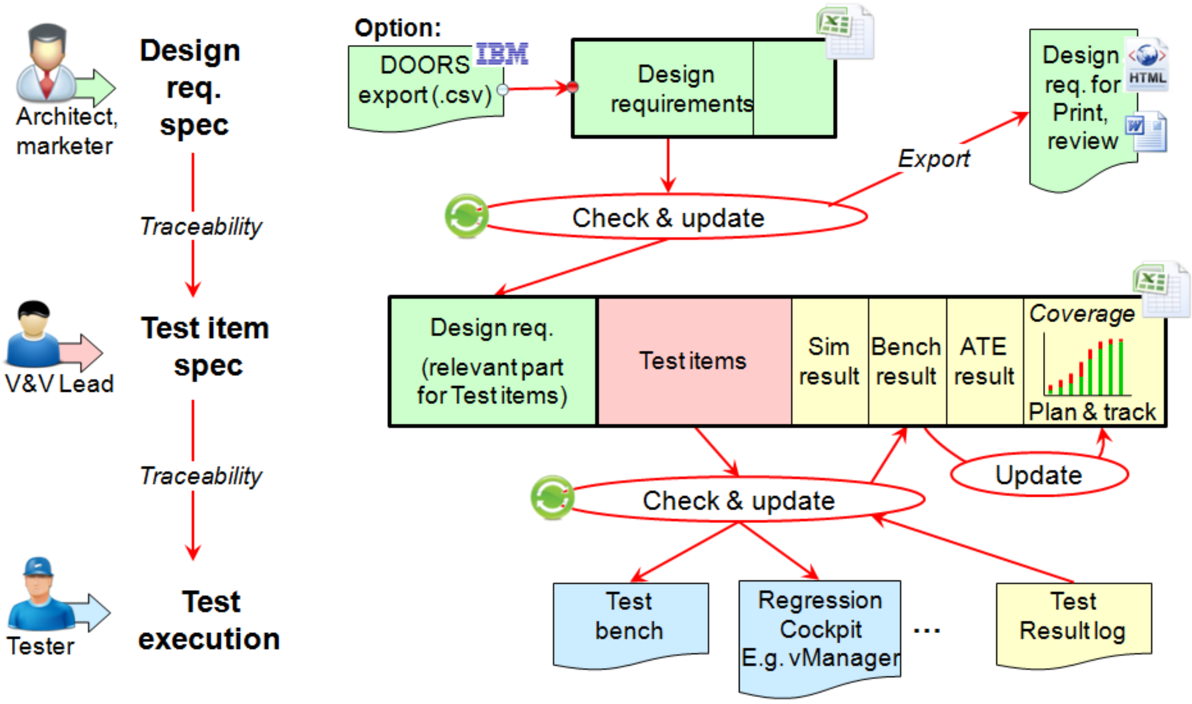

Figure 2 In the V&V traceability flow, the test item spec is the central part

The test item specification covers multiple V&V domains, e.g. functional verification (part of the first "V" of V&V), Bench (Validation in lab using real silicon), PQA (Product Quality Assurance) and ATE (Automated Test Environment – production testing). The strength of this flow is the support to reuse test items over these multiple V&V domains. The test item spec indicates for each test item in which domain it must be applied. For each test item, the corresponding owners should together agree, before that test item can get the 'approved' status. As soon as all test results are back-annotated in the test item spec, the results of these different domains can be easily compared. Discrepancies are easily detected, highly improving the overall V&V quality.

Figure 2 also shows the capability to export the specification data to HTML format. The Check & update automation assumes that every item (requirement, test item) is stored in one Excel row. Especially when multiple domains are supported, the test item row can have a lot of columns. Especially when the specification is being displayed on a beamer for group review purposes, sliding horizontally through the columns is impractical. Because the Check & update tooling is fully aware of the spec items and their attributes, it can easily generate (on the fly) a human readable table. Each (test) item is formatted into an HTML table, which is printer and beamer friendly and fully configurable (headers, font size, colors, etc.) to the specific occasion. The Excel file remains the “golden reference” specification; the HTML files are only generated views from this single source.

In this flow, the test item spec acts as a verification cockpit. The proper overview supports planning-and-tracking of test coverage over time. E.g. the amount of Passed/Failed/non-implemented test items can be stored in a table at a weekly basis. From this table, Excel can easily create graphs to track the process and closure. In order to improve the planning using the table, the test items can have a simple weight attribute (e.g. ranging from 1 to 3), to roughly indicate test item implementation time.

B. Check & update mechanism

Because design requirement spec and test item spec are maintained by different owners concurrently, at some point of time the shared attributes in the documents may get inconsistent. It doesn’t help to combine both levels of information in the same document, because then the shared requirement (green) attributes are consistent, but in the test items spec the updated requirement attributes are inconsistent with the (older) test item attributes (red). In order to allow concurrent changes in requirements and test items, these should be stored in separate files.

Next to that, an explicit synchronization is needed between the separated documents. This means a handshake between the architect and V&V lead, at a moment that the V&V lead has time to process the inconsistencies, one-by-one. So, for every inconsistent requirement the corresponding attributes in the test item(s) are updated by the tool, only if the corresponding test items attributes are updated (manually) too. The tooling is helping to clearly trigger the requirement inconsistencies.

Next to this Check-one-by-one, there is a Check-all feature, which gives a quick overview of all inconsistencies. In addition there is an Update-one-by-one and Update-all, which performs the updates one-by-one or in one go, respectively. This update mechanism is to be used in the initial stage only. Next to these actions there is also an Add-all-missing action, which will generate items from the (higher) level. E.g. for each design requirement we expect at least one test item, the missing items can be generated with only the proper design requirement attributes.

The Check & update mechanism is possible between two Excel files, between Excel and text files or between Excel and XML files. The Check & update is performed by all the owners of the V&V traceability flow. In a project, ownership of the attributes (corresponding to the various columns in Excel) is clearly defined in this flow.

The update rules for these attributes are specified in configurations (modes) of the Check & update tool, which will be explained in the next section. The configuration modes end up in tabs of the Check & update graphical user interface.

C. Configurability of the update mechanism

In order to get acceptance of such traceability mechanism, the tooling should not enforce specific templates because of traceability automation, but it should adapt to templates being used. The test item spec has the challenging objective to cover the large part of V&V spectrum. As discussed before, it can include multiple V&V domains, e.g. functional verification, Bench, PQA and ATE. So in the Check and update tooling much effort is spend on configuration capabilities, to support different templates.

In the Excel files spec, each item is stored in an Excel row. Each spec item has attributes, using the columns in the Excel worksheet. In order to synchronize two columns, the columns are being identified by a unique label. In Figure 3 the column labels are shown in row two of both Excels. In a configuration file, each column which needs to be synchronized is specified, including the direction (top down / bottom up) and the folding function in case of making updates bottom-up (from test results to test items).

A dedicated column (and label) is introduced to specify the purpose of the row. So we can e.g. distinguish headers and information rows from spec items. The traceability is only applied on the spec items (e.g. requirements, test items, test results). To link spec items of two Excel files, each spec item requires an identifier. The (sorted) order of the spec items doesn't have impact on the traceability between the Excel files.

The implemented configuration mechanism is split into a project-wide and user-specific part. The most obvious user-specific configuration is specifying the owner value in the test item file, causing only items belonging to that user are being synchronized during the Check & update.

IV. PROOF OF CONCEPT

This section demonstrates the V&V traceability flow. Figure 3 shows the implemented Check & update mechanism between test items and parametric simulation data. The Parametric Excel is the interface of an existing parametric simulation environment used in NXP.

The test item Excel in this figure shows the test item specification, which contains attributes of the requirement specification (green), and derived from that are the test item specification, e.g. the min/max spec values (red) under different conditions (PVT corners: Process, Voltage and Temperature values). The min/max spec values of each parameter in the test item Excel are synchronized with the parametric Excel. The Check & update mechanism for these spec items is done in a top-down manner, where agreed updates will update the Parametric Excel using data from the test item Excel. The parametric simulation results are of course updated in a bottom-up matter. Multiple simulation results are grouped to a single test item result, using a specific fold function (e.g. min, max, pass/fail, collect).

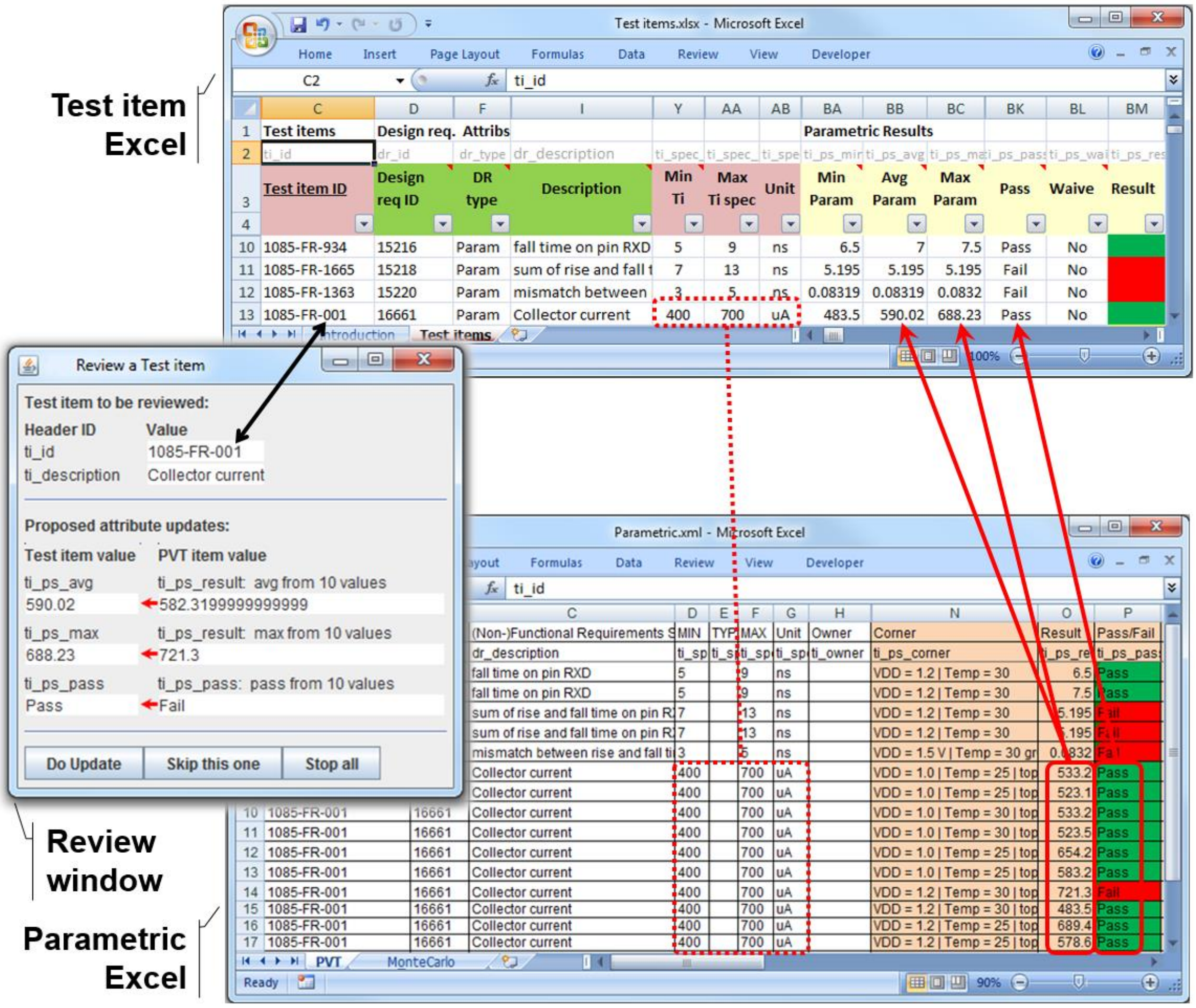

Figure 3 Check & update between test items and parametric simulation

Configuration of each column being synchronized is done via a project-wide configuration file. Normally multiple testers are working concurrently, using the same test item spec, but each having their own Excel for parametric simulations, with detailed information.

In Figure 3, the review window of the check and update tool is shown. As part of the check & update process, attributes can be updated by reviewing the test items one-by-one. The upper part in the Review window shows a configurable number of attributes describing the test item. The lower part shows the inconsistencies including the direction of the potential update. The main program window (not shown) gives more detailed logging about the cell references and folding functions being used.

The whole V&V traceability flow has been applied into two projects within NXP. An architect has written design requirements in the Excel template, proposed by this work. The V&V lead has generated from these requirements an initial set of test items in a test item excel, as in Figure 3. Finally the architect and V&V lead use the same tool to keep their Excel files synchronized. The test items again are being synchronized by the testers, each having their own Parametric Excel. Each owner knows exactly the attributes he can change and the attributes are obtained from the other owners. Specifying clearly the roles and responsibilities is important ingredient for successful V&V flow.

In this case, the test item specification is fully traceable from design requirement to the large set of simulation results from parametric simulation. Test coverage was properly tracked using the test item spec. The test item spec has still reasonable size capturing only a summary of the parametric results, as shown Figure 3. These results are easily traceable to the verbose parametric data stored in multiple parametric Excel files owned by different testers.

A. Future work

The department running the projects has decided to roll-out this approach to all their projects. Next to that, we apply this approach to synchronize test item data in test implementation files for the functional simulation domain. For a different department we started to synchronize large Excel files containing many validation measurements with the test item spec containing a proper summarized overview.

CONCLUSIONS

The presented paper introduces a V&V traceability flow, supported by tooling, enabling traceability and consistency of the design requirements, test items, and test results. A check & update mechanism is implemented in a stand-alone tool, to safeguard the consistency throughout the flow. The test items spec contains the result summary of the test execution, which is the proper overview to track the test coverage during the project. Different owners in the V&V flow (architect, V&V lead, and testers) are now capable to use this tool to keep all V&V items consistent.

The proposed V&V traceability flow is successfully applied in two NXP projects, which included traceability between design requirements, test items and parametric simulation. This resulted in a more effective and efficient V&V process and increased product quality. Now the V&V traceability flow is being applied within NXP to other V&V domains such as functional verification.

ACKNOWLEDGMENT

We thank a number of NXP colleagues: Pieter Hooijmans for being the initiator and first sponsor of this work, Frans Theeuwen for given the opportunity to work on this topic for multiple years, Maikel van Beurden for the cooperation on parametric simulation, Stephen van den Elshout and Jan-Maarten Schenk for evaluating this flow in their projects, and all others showing interest in having a structured V&V flow.

REFERENCES

[1] Yufei Xie, Tao Tang, Tianhua Xu, Lin Zhao, "Research on requirement management for complex systems", Computer Engineering and Technology (ICCET), 2nd International Conference, 2010, pp. V1-113 - V1-116.

[2] IBM® Rational® DOORS® is a requirements management application for requirements communication, collaboration and verification www-03.ibm.com/software/products/en/ratidoor.

[3] TestLink is a web based Test Management tool. sourceforge.net/projects/testlink/ testlink.org

[4] Cadence® Incisive® vManager™ is the solution of Cadence for verification planning and management www.cadence.com/products/fv/vmanager/pages/default.aspx

[5] www.magillem.com/eda

[6] www.magillem.com/doc