

# Understanding DO-254 Compliance for the Verification of Airborne Digital Hardware

October 2009

## Authors Abstract

**Dr. Paul Marriott**  
 XtremeEDA  
 Corporation

**Anthony D. Stone**  
 Synopsys, Inc

This whitepaper is designed to provide a basic understanding of the main concepts of the DO-254 compliance specification for electronic component design. It outlines the major steps involved in a DO-254 compliant ASIC/FPGA design and verification process, and explains how differentiating tool features can be mapped to enhance and facilitate critical stages of the DO-254 process.

## Introduction

As the amount and complexity of electronic content has grown in commercial aircraft, it became necessary for the FAA to establish a baseline of minimum design flow steps for airborne equipment. DO-254 was formally recognized in 2005 as a standard for ensuring the highest level of safety in electronic airborne systems. It includes five levels of compliance, known as Design Assurance Levels (DAL), that range in severity from A (where hardware failure would result in catastrophic failure of an aircraft) to E (where failure would not affect safety). As expected, meeting a “DAL A” level of compliance requires significantly more effort and greater attention to verification than would “DAL E”.

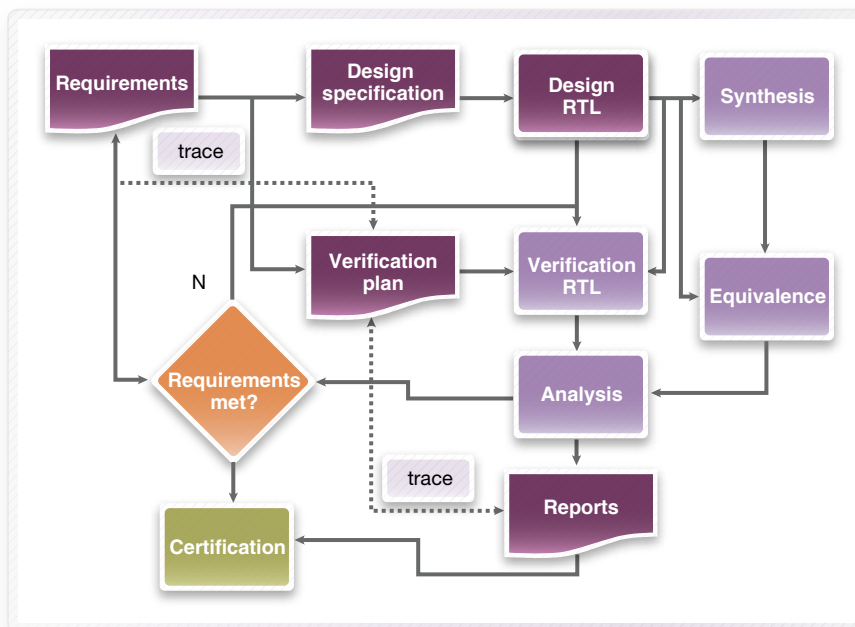


Figure 1: Typical DO-254 design flow steps

Compliance to this standard involves a process that is more rigorous than the standard ASIC/FPGA design and verification flow. While the tools used to design and verify the hardware are the same as in non-DO-254 applications, the process involves additional steps, particularly in the area of functional verification.

### Parts of a compliant flow

DO-254 applies to complex airborne hardware, such as ASICs, FPGAs, and PLDs. According to the specification, a hardware item is considered “complex” if a comprehensive combination of deterministic tests and analyses cannot ensure correct functional performance under all foreseeable operating conditions. For complex devices, a rigorous, structured design and verification process takes the place of exhaustive testing. Demonstrating that the development and verification of complex hardware complies with this process is the objective of DO-254.

A DO-254 compliant design is specified using a set of formal requirements. As part of the certification process, the applicant must prove that their implementation meets all of these requirements. A graphical illustration of the typical process flow is shown in Figure 2.

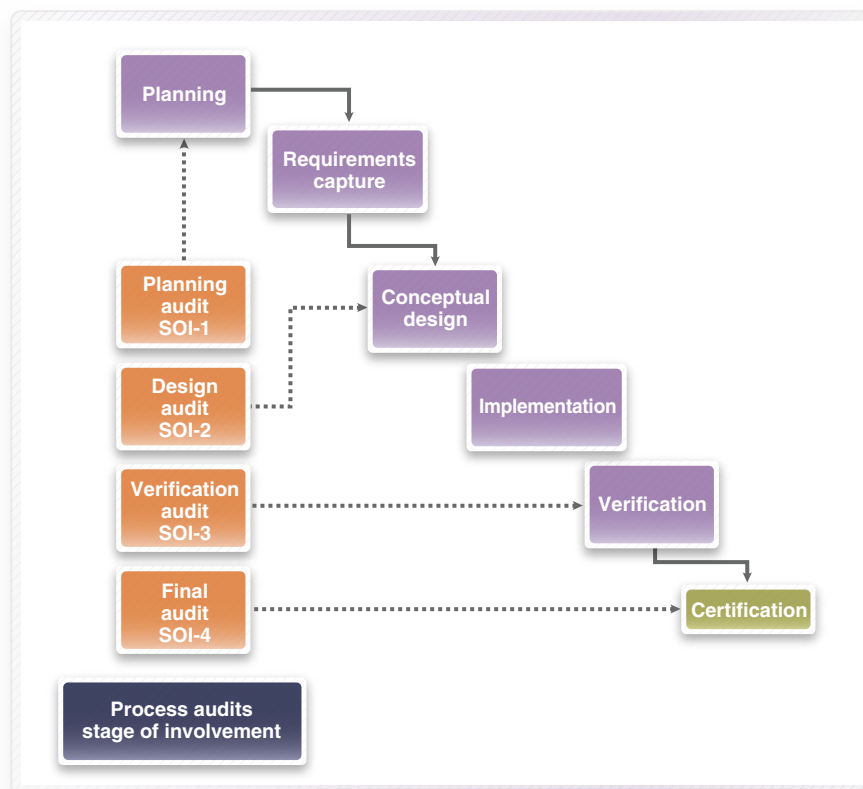


Figure 2: DO-254 process flow

Note that the process described in this document is a template. Depending on the nature of the project, different stages and/or tools may be required. A key principle, however, remains constant for successful DO-254 compliant flows: verification results (simulation waveforms, regression status, coverage figures) must be traceable and linked to the formal requirements. The process of traceability may be either automated or manual; the output capabilities of the tools utilized in the flow will determine both the ease and degree of automation.

### Important points to note about DO-254

DO-254 is, first and foremost, a process specification; it does not specify the detailed implementation of that process. In particular, a very common misconception exists that there are “DO-254 Certified” tools available – this is not the case as it is the overall process itself that receives certification, and not individual elements

such as the tools. The certification of the process is the demonstration that the specified process was actually performed in a verifiable and repeatable manner to the highest level of confidence.

A DO-254 compliant design is, hence, specified using a set of formal requirements. As part of the certification process, the applicant must prove that their implementation meets all of these initial requirements. Typically this is achieved through functional verification, which must prove adherence to the formal requirements- not unlike the best practices employed in the majority of hardware development flows in use today.

## Tool assessment

Tool assessment is a part of the DO-254 process that is meant to ensure that the tools used for hardware design and verification perform correctly; those chosen must be, of course, suitable for their intended tasks and specified so that the process is traceable and repeatable. For example, a synthesis tool takes an RTL design and converts it into a gate-level netlist. To ensure that the output of this tool is to be trusted, some justification for this assertion is usually required. This is where tool assessment comes into play.

Identifying the process the tool supports involves describing the purpose of the tool and the role of the tool in the project. A particular document, the Plan for Hardware Aspects of Certification (PHAC), is crucial to the DO-254 process, and should contain this information, as well as any known limitations (bugs, errata) of the tool. Below is an example of how an applicant might describe their use of VCS as a verification tool:

### *VCS Usage*

*This tool supports the process of simulating and debugging ASIC designs at both the RTL and the gate level. It will be used to simulate Verilog RTL code at both the block level and at the top level of the ASIC. The coverage features of VCS will be used to assess the completeness of the verification effort. Once the design has been synthesized, VCS will be used for gate level simulation to ensure that the results are the same as those obtained during RTL simulation.*

For applications classified as “verification tools”, tool assessment is only required for highly critical designs; in DO-254 parlance these are denoted as Design Assurance Levels (or “DAL”) A and B.

The first steps in the tool assessment process are to identify the tool(s) used and the processes they support. Tool identification requires specifying the name of the tool, the source, version, and host environment where it is running. Below, for example, is sample identification for VCS running under Linux:

```
Name: VCS  
Source: /tools/synopsys/vcs/C-2009.06/bin/vcs  
Version: VCS-MX C-2009.06-B-3  
Host Environment: Dell PowerEdge 2900 x86 64 bit server,  
running Red Hat Enterprise Linux ES release 4
```

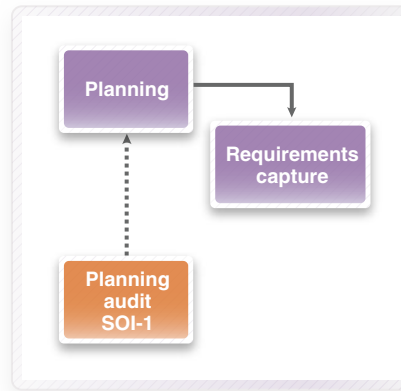
Once the identification steps are complete, the remainder of the tool assessment process can take one of three forms, depending on particulars of the applicant and the project:

1. **Independent Output Assessment:** Can the output of the tool be verified to be correct through an independent means? Some possibilities include manual review of tool output, comparison with a second equivalent tool (e.g. in the case of VCS, another simulator), and comparison between RTL/gate simulation and the actual device (FPGA).
2. **Relevant History:** Does the tool have a well-documented history of usage where it has consistently produced acceptable results? The history of usage may include airborne and non-airborne applications.
3. **Basic Tool Qualification:** This is a process requiring several stages and culminating in a report known as the Tool Qualification Accomplishments Summary (TQAS). Tool qualification should only be undertaken if the previous two methods are not applicable.

This choice should be discussed with the Designated Engineering Representative (or “DER”, an appointed engineering resource who has the gating authority to judge DO-254 compliance) early in the development process to ensure that it is acceptable. The method should, as well, be described in the PHAC document. For more information on tool assessment, please refer to the DO-254 specification .

### Requirements capture, planning and traceability

DO-254 stipulates that a design must be specified using formal requirements. The first step in any DO-254 development is to capture these requirements, often using a special-purpose tool, such as IBM’s (Telelogic) DOORS. Requirements are written hierarchically, meaning that a high level requirement may be composed of several simpler ones. Demonstrating that all the lower level requirements have been satisfied proves that parent requirement has been met as well.



From these captured requirements, two documents are derived: a design specification, and a verification plan. Both of these are typically implemented as text documents; however the use of a focused plan tracking application offers an alternate means of creating a verification plan (see section 4).

Another important product of the planning phase is the aforementioned PHAC document. According to section 10.1.1 of the DO-254 specification, “The PHAC defines the processes, procedures, methods, and standards to be used to achieve the objectives of this document and obtain certification authority approval for certification of the system containing hardware items.” The PHAC may refer to the verification plan and describe the methodology used to achieve total verification (i.e. constrained-random, coverage driven, directed test, assertion-based) and explain the traceability mechanism in place. The PHAC should also specify the EDA tools to be used in the project, and outline the method of tool assessment for each.

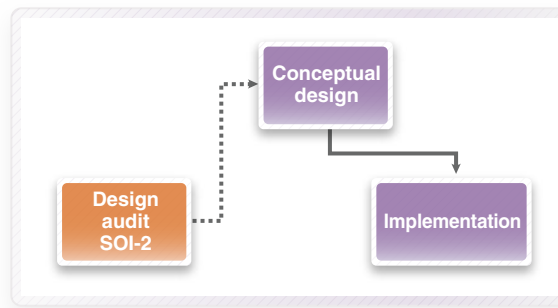
Configuration management is another important consideration during the planning stage. For DO-254 compliance it is important to control the version and history of all “artifacts” associated with the project. In DO-254 terms, an artifact is anything that is generated as a product of the ASIC or FPGA design and verification process. This includes all documentation, RTL and behavioral code, scripts, and reports. EDA tools should also be revision controlled in order to satisfy the tool assessment requirements. The configuration management strategy should be put in place in advance of any development work, and should be described in the PHAC.

After all documents are completed, an audit of the planning stage is initiated with a DER (Stage of Involvement 1) before the implementation stage begins.

### DO-254 Compliant Implementation

Once the planning process is complete, the process of design implementation begins. Designers of ASIC and FPGAs alike will start with a concept, and develop RTL code, using assertions along the way to check the functionality vs. the set requirements. At this stage tools for design entry, waveform analysis, and RTL and assertion debug are highly useful. For FPGA-focused development, this should include the incremental design and debugging of source code.

Code coverage is necessary for certification to levels A and B; every line of code must originate from the implementation of one or more formal requirements as well as be stimulated in the verification process. In addition, the output of the synthesis tools must also be verified.



The DO-254 standard does not specify or dictate the choice of tools or methodologies- it, instead, requires that the final design implements the formal requirements. However, in order to comply with DO-254 standards (as well as the ultimate success of the design), implementation tools should be chosen that have shown demonstrated success in similar projects. Justification for a tool can include it either having been accepted for internal use previously, being cited as an industry standard, or having been widely adopted throughout the broader design community.

### **RTL implementation for ASIC and FPGA**

The main implementation tool of either an ASIC or FPGA design is that of synthesis, preferably with the ability to optimize timing, area and power consumption. For ASIC designs, Synopsys IC Compiler provides the necessary synthesis functionality combined with advanced features in routing, timing, and Design-for-Test that enhance the predictability, quality and reliability of an ASIC design. For FPGA/PLD devices, the combination of Synplify® Family (for FPGA synthesis) and Identify debugger (for debugging with high internal visibility) provides the capability to perform RTL synthesis to widely used PLD devices, at-speed debugging of FPGA designs and incremental design and debugging of source code. A schematic creation capability is also included, which helps to fulfill any potential design audit requirements.

In either ASIC or FPGA implementation, RTL is the input to these tools; the same code is also used to assess the output of the synthesis tools independently as part of the DO-254 process assurance. This code is also verified by simulation and the necessary independent output assessment of the synthesis tools is performed by gate level simulation or by formal equivalence checking tools. Once the RTL is stable, the applicant should undertake a design audit with the DER (Stage of Involvement 2) to ensure that the design process is acceptable and meets the objectives specified in the planning documents.

### **Analog and mixed signal simulation for DO-254?**

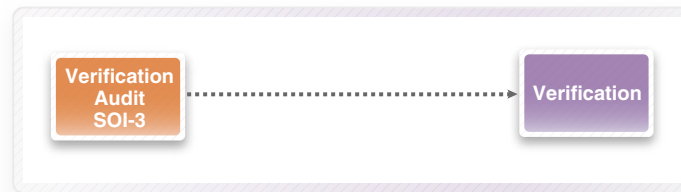
DO-254 is primarily concerned with “micro-coded digital hardware”, but, ultimately, all designs are essentially analog implementations in physical transistors. As a part of the overall implementation and verification loop for DO-254 compliance, it is important that equivalent representations of circuit blocks yield the same functional results regardless of their level of abstraction.

Synopsys’ SABER provides mechatronic, mixed-signal simulation capabilities, as well as detailed analysis of performance versus time and frequency domains and versus thermal constraints. It provides a large library of mechanical, electrical, and magnetic components, allowing designers to characterize airborne hardware under real-world operating conditions. Its co-simulation capability with Verilog (via the VCS simulation engine) permits the designer to develop and implement multiple, equivalent representations of individual blocks as needed.

Since timing requirements are encapsulated as formal requirements, the ability to verify these prior to tape out in a productive environment is highly important to achieving a final working product. CustomSim and HSPICE simulation solutions can be used to verify that the post-layout implementation of the design is equivalent to the gate-level netlist, the most accurate means of performing timing qualification on the design and assuring functional equivalence across multiple design representations.

## Verification in a DO-254 Compliant Flow

Verification is the most crucial phase to the DO-254 process since this is where the design is checked vs. the initial formal requirements. As such, an efficient mechanism to link the verification plan and its results back to the original requirements documents is highly valuable for demonstrating DO-254 compliance (not to mention, of course, building a successful end product).



Once the verification effort has been deemed complete, the applicant must undergo a verification audit with a DER (Stage of Involvement 3), including a review of the reports and documentation compiled during the verification process, as well as test procedures, results, and coverage analysis.

### Directed testing methodology versus constrained-random approaches

The format of the requirements documents is such that many verification teams pursue a directed testing approach to verification- writing and executing through simulation one test per requirement (or per each set of closely related requirements). The verification plan document should contain a description of each test and which requirement(s) it verifies. The document should also spell out goals for code and assertion coverage (at minimum, 100% statement coverage is required by DO-254).

Verification is complete when every requirement has been exercised and, in the case of level A and level B designs, code coverage is complete –all features have been implemented in the design and stimulated in the verification process.

As the directed tests are written, the process of simulation, regression testing, and debug begins. Simulation and code coverage analysis may reveal shortcomings in the directed test model. It is often necessary to add new tests to address untested code and corner cases. The verification plan should be updated accordingly to describe the purpose of each test and the requirement(s) it verifies. Similarly, care must be taken to adjust the verification plan and directed tests any time there is a change to one or more requirements.

The bulk of modern electronic designs are sufficiently complex that complete verification of all functional corners using a directed test approach is impractical, and a constrained random verification methodology is warranted. Using this approach requires the implementation of a sophisticated, self-checking verification environment which can predict and verify the behavior of the device under test under all conditions. In this methodology, there is not a one-to-one correspondence between the formal requirements and the associated tests. Instead, features in the test environment, such as assertions and functional coverage, are used to track verification completeness. This strategy is often used when the design being verified is very complex, and creating individual, directed tests for every feature would become onerous. The verification environment itself still encapsulates the correctness criteria for the set of features being verified, but individual simulation runs may not necessarily cover all features. Running many tests with different random seeds (and maybe different sets of constraints) will result in coverage information being gathered, and then aggregated across all tests until 100% coverage is achieved.

In order to maximize productivity in such an environment, a structured mechanism that links the coverage results back to the formal requirements is necessary in order to demonstrate the completeness of the verification. Correctness and test completion criteria are still required in order to determine that the tests themselves have passed or failed. Verification cannot be considered complete until all tests are passing with 100% coverage (both functional and assertion) AND 100% code coverage is achieved. Automated applications that enable the systematic planning and tracking of verification progress are available, adding significant value to a DO-254 flow by easing the tracing and documentation of the verification phase.

Customers may choose not to pursue a coverage driven, constrained-random verification methodology for a variety of reasons. Perhaps the design complexity does not warrant it, or the tool set being used does not support it. In this case the same verification process must still take place, but more manual effort is required.

### **Formal verification and equivalence checking**

As part of the verification process, formal verification (model checking) offers additional benefits in a DO-254 environment. Assertions may be difficult to prove or disprove because they monitor a behavior that occurs over a long period of time, or specify a condition that should never occur under normal device operation. Applications that provide a formal, mathematical means of exhaustively verifying the behavior described by assertions add value to DO-254 compliant verification by demonstrating that assertions describing erroneous behavior are indeed unreachable. They can also prove that behavioral rules defined in assertions hold true for all possible design states. Since assertions are derived from formal requirements, the requirement itself is proven through the use of formal checking.

As most verification is performed on RTL, DO-254 requires that the applicant demonstrate that their gate level netlist is equivalent to the RTL implementation. For smaller designs, gate-level simulation may be sufficient; larger, more complex designs require a more complete approach- equivalence checking. This can be achieved via either formal, static verification methods and does not require the often tedious, error-prone development of test vectors, or through functional comparison of the design in progress vs. a reference design (e.g. an RTL view vs. a SPICE netlist view).

If the formal requirements require verification of clock domain issues, accurate verification of synchronization of multiple clocks and of the consistency of design and delay calculation constraints will be necessary; detailed timing simulations may also be run to provide an additional layer of verification.

### **DO-254 Verification Planning**

A hierarchical verification plan, or HVP, is a natural way of representing requirements and associating verification metrics with each. Use of an HVP mark-up language also facilitates traceability during the verification stages. The verification plan itself can take the form of either a Microsoft Word document or an Excel spreadsheet, depending on the needs and preferences of the customer. One advantage of the spreadsheet format is the ability to exchange information with DOORS.

For DO-254 customers designing ASICs and complex FPGAs, this is the preferred verification methodology. A coverage driven, constrained-random verification process begins with a plan that associates design features with functional coverage groups and assertions. Powerful planning tools are available that can be used to create an HVP that is directly derived from formal requirements, as well as associating verification criteria with each requirement. Figure 3 below shows possible output after exporting requirements defined in DOORS into a Microsoft Excel file.

ID	Object Level	REQ	ProductVersion	Req Level	Req?	Verification
REQXX001	5	The Local Processor Address Bus interface shall support a 36 bit wide address bus in accordance with the PowerPC Specification section 2.3 'Address Transfer Signals' and 3.2.2 'Address Transfer'.	ALL	Parent	TRUE	Simulation
REQXX002	5	The module shall accept the Transfer start signal from the processor in accordance with the PowerPC Specification section 2.2 'Address Transfer Start Signals'.	ALL	Parent	TRUE	Simulation
REQXX003	5	The module shall accept the Transfer Type signals from the processor in accordance with the PowerPC Specification section 2.4 'Address Transfer Attribute Signals'. [Commentary: All Transfer Types are supported except the Reserved types.]	ALL	Parent	TRUE	Simulation

Figure 3: Requirements exported from DOORS into Excel

Figure 4 shows the contents of the Excel file converted into an HVP, the format required by Synopsys' VMM Planner. Coverage or assertion metrics have been associated with each requirement.

hvp plan	PowerPC interface	feature	subfeature	Sdescription	value ppc_if_Group	value ppc_if_Assert	value ppc_if_Line	measure ppc_if_source
plan	PowerPC Interface							
			RCQXX001	The Local Processor Address Bus interface shall support a 36 bit wide address bus in accordance with the PowerPC Specification section 2.3 'Address Transfer Signals' and 3.2.2 'Address Transfer'.				group instance ppc_if.address_tenure
			RCQXX002	The module shall accept the Transfer start (TSF) signal from the processor in accordance with the PowerPC Specification section 2.2 'Address Transfer Start Signals'.				property ppc_if.assert_transfer_start
			RCQXX003	The module shall accept the Transfer Type signals from the processor in accordance with the PowerPC Specification section 2.4 'Address Transfer Attribute Signals'. [Commentary: All Transfer Types are supported except the Reserved types.]				group instance ppc_if.transfer_type

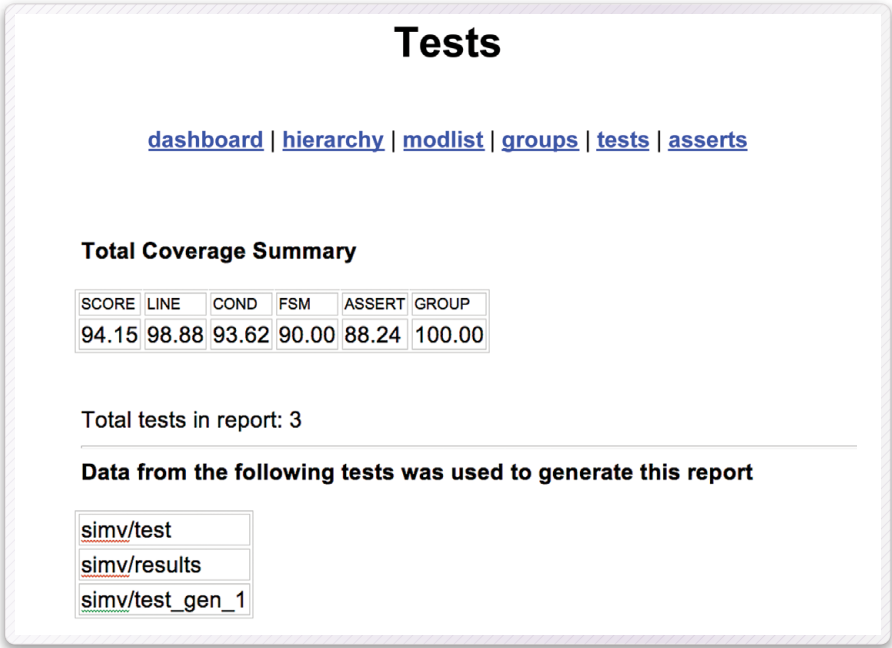
Figure 4: Requirements in an HVP spreadsheet

Constrained random, coverage-driven verification also requires the implementation of a sophisticated, self-checking verification environment. The combination of a verification planner based methodology and verification IP provides a proven means of building a DO-254 compliant verification environment faster and with greater confidence vs. other methods.

Once the environment is complete, the process of regression testing begins, with the goal of achieving the verification objectives specified in the HVP. Comprehensive reports are needed to show coverage progress, and the tests and random seeds that were used to attain it. Once the coverage goals have been achieved and verification is complete, the reports and annotated verification plan are the evidence that a DER will require to certify successful DO-254 compliance.

Traceability is critical for DO-254 compliance, and reports are needed to prove to a DER that verification of the design has been complete. As shown in Figure 5, a report generator (such Synopsys' URG) can generate reports showing code and assertion coverage statistics, and indicate which tests contributed to this coverage. This data should be an integral part of the PHAC document, as it is an artifact that shows the completeness of verification versus the initial coverage plan.

If an integrated URG capability is not available, customers may either use standalone third party tools or implement (and support) their own reporting system.

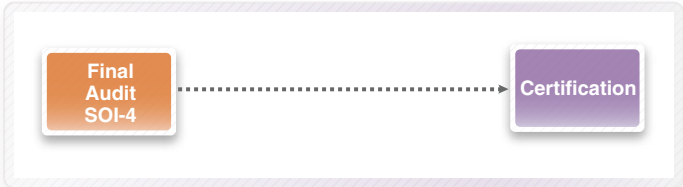


**Figure 5: Coverage report generated by URG**

The other important aspect of traceability is the linking of the regression results, coverage numbers and the original requirements. This can be done in the verification plan using a table which cross-references test status and coverage to requirements. Some customers may also choose to manually enter this information into their requirements capture tool (e.g. DOORS). This information must be made available to the DER during the verification audit.

## Conclusion

The Hardware Accomplishments Summary (HAS) is a document which assembles the compliance and completion data accumulated during the project. This document should demonstrate that the objectives outlined in the PHAC have been achieved. Review of the HAS is part of the final audit which takes place with the DER at the end of the development process (Stage of Involvement 4). Once the DER is satisfied that a compliant process has been followed, they can submit the appropriate paperwork to the FAA for certification.



Synopsys provides a complete portfolio of tools and technologies that enable the quick assembly of a productive DO-254 compliant design and implementation flow. This formidable lineup includes industry-leading functional verification, production-proven equivalence checking, and powerful FPGA synthesis and debugging solutions. More importantly, it includes the capability to track, link and report verification results vs. initial requirements - a key to proving DO-254 flow compliance. As DO-254 designs increase in complexity and size, Synopsys solutions and methodologies are available to help engineers achieve high quality, DO-254 compliant hardware, quickly.

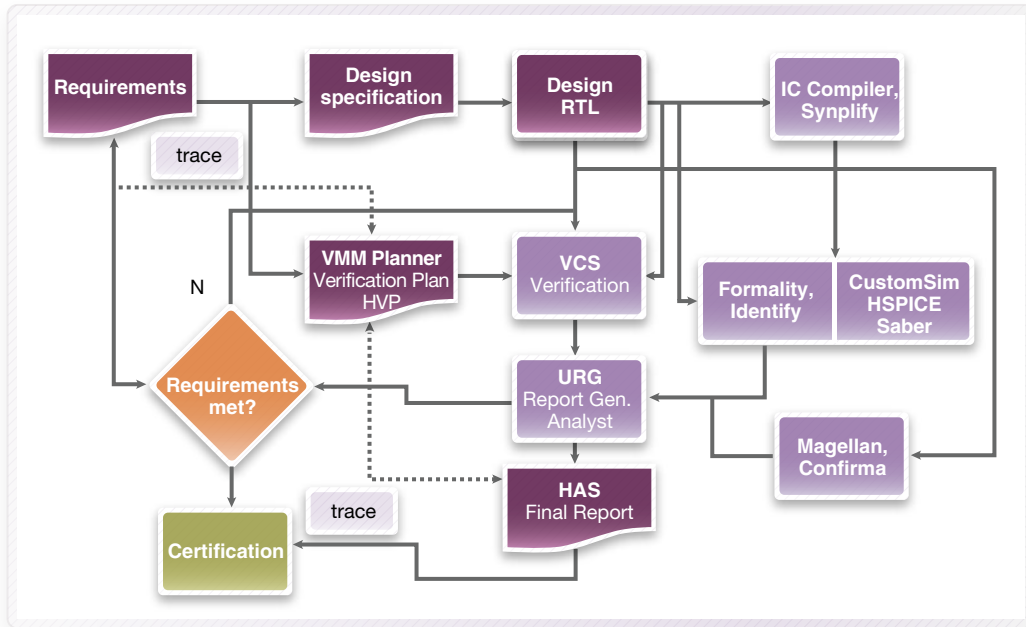


Figure 6: DO-254 Design Flow with Synopsys Tools

## Resources

For further information, visit the DO-254 User's Group at <http://www.do-254.org>. A copy of the specification itself can be purchased from the RTCA organization at <http://www.rtca.org/onlinecart/product.cfm?id=194>.

Newcomers to the DO-254 process may benefit from training and through the engagement of consultants. Skilled in advanced verification techniques and familiar with the DO-254 process, consultants can provide the head start needed to achieve maximum productivity and project success.