

# UNDERSTANDING DO-254 AND SOLUTIONS TO FACILITATE COMPLIANCE

MICHELLE LANGE  
MENTOR GRAPHICS DO-254 PROGRAM MANAGER



W H I T E P A P E R

[www.mentor.com](http://www.mentor.com)

## INTRODUCTION

RTCA/DO-254, or simply DO-254 (also known as ED-80 in Europe), is a document developed under the guidance of the RTCA ([www.rtca.org](http://www.rtca.org)) that establishes “Design Assurance Guidance for Airborne Electronic Hardware.” As hardware designs get more and more complex, and virtually impossible to exhaustively test, the purpose of DO-254 is to establish a framework for the development of such hardware to ensure, by way of a standard framework, that the resulting hardware devices perform their intended function under all foreseeable conditions.

The Federal Aviation Administration (FAA), European Aviation Safety Agency (EASA) and other worldwide certification authorities have invoked DO-254, making it policy for the development of “custom micro-coded components” (i.e., PLD, FPGA, and ASIC devices). This has led to its widespread adoption in commercial aviation programs in recent years. Some airframe integrators are also mandating compliance beyond the component-level. DO-254 compliance is also becoming increasingly common on military projects.

As is almost always the case with relatively new standards, initial efforts to comply with DO-254 can be fraught with difficulty and unexpected costs. Fortunately many of these challenges can be minimized with a thorough understanding of what DO-254 compliance really entails, and using work-flows and solutions supportive of this.

## DO-254 OVERVIEW AND EVOLUTION

---

### WHY COMPLY WITH DO-254?

DO-254 was originally developed to serve as a means of compliance to the Code of Federal Regulations (CFRs) Title 14 “Aeronautics and Space” for parts 23 (normal, utility, acrobatic, and commuter category airplanes), 25 (transport category), 27 (category rotorcraft), 29 (transport category rotorcraft), and 33 (aircraft engines). Within the CFR, each of these parts has subparts .1301 and .1309, which are pertinent in the discussion of DO-254 compliance.

Part.1301 “Function and Installation” says “Each item of installed equipment must:

- (a) Be of a kind and design appropriate to its intended function;
- (b) Be labeled as to its identification, function, or operating limitations, or any applicable combination of these factors;
- (c) Be installed according to limitations specified for that equipment; and
- (d) Function properly when installed.”

Likewise, Part.1309 “Equipment, Systems, and Installations” says:

- (a) The equipment, systems, and installations whose functioning is required by this subchapter, must be designed to ensure that they perform their intended functions under any foreseeable operating condition.
- (b) The airplane systems and associated components, considered separately and in relation to other systems, must be designed so that--
  - (1) The occurrence of any failure condition which would prevent the continued safe flight and landing of the airplane is extremely improbable, and
  - (2) The occurrence of any other failure condition which would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions is improbable....”

Thus, complying with DO-254 means complying with the CFRs. Countries outside the United States have similar sets of regulations that are likewise satisfied by compliance to this document.

### WHEN DID IT BECOME POLICY?

DO-254 was written to apply to all levels of hardware design and finalized as an industry document in 2000. Eventually, the FAA invoked DO-254 as policy with Advisory Circular (AC) 20-152, which changed the scoping. This policy document, issued on June 30, 2005, states that “This AC recognizes the guidance in RTCA/ DO-254 applies specifically to complex custom micro-coded components with hardware design assurance levels of A, B, and C, such as ASICs, PLDs, and FPGAs.”

Prior to this, the FAA and airframe integrators had used compliance to similar requirements that were defined by special documents such as Issue Papers (IPs) or Certification Review Items (CRIs) for specific aircraft programs. In some cases this meant following DO-178B, which is the complementary standard for software design assurance. Today IPs and CRIs still do influence how DO-254 must be complied with on aircraft programs.

The interpretation of DO-254 has been clarified and influenced by two FAA documents: Order 8110-105 and the “Conducting Airborne Electronic Hardware Design Reviews,” otherwise known as the Job Aid. Order 8110-105 clarifies a number of ambiguities in the original invocation of DO-254 from AC 20-152. The Job Aid is an internal FAA document used by DO-254 auditors as they perform their “Certification Liaison” tasks. Both of the documents are public and can be found on the FAA web site ([www.faa.gov](http://www.faa.gov)).

Figure 1 shows DO-254 as a means of compliance to the CFRs, along with the other types of policy documents that influence its interpretation.

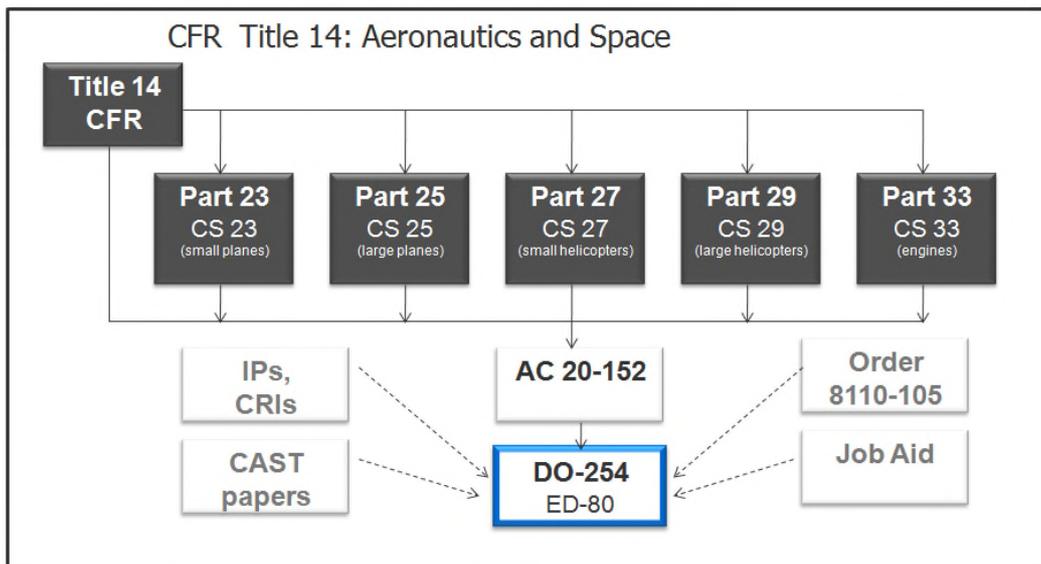


Figure 1. Regulation, DO-254, and Related Policy

It’s important to have a big picture understanding of why DO-254 compliance is necessary and all of the related policy documents and regulations that influence its interpretation. Thus, complying with DO-254 is not as simple as reading the document and following what it says. The document itself was written with a different scoping than it is currently applied, and subsequent policy documents and opinions have influenced its interpretation and today’s common compliance practices.

## UNDERSTANDING THE DO-254 LIFECYCLE

The best summary of the DO-254 document itself, how to navigate it, and what compliance entails is found in Figure 5-1 of DO-254 (page 28).

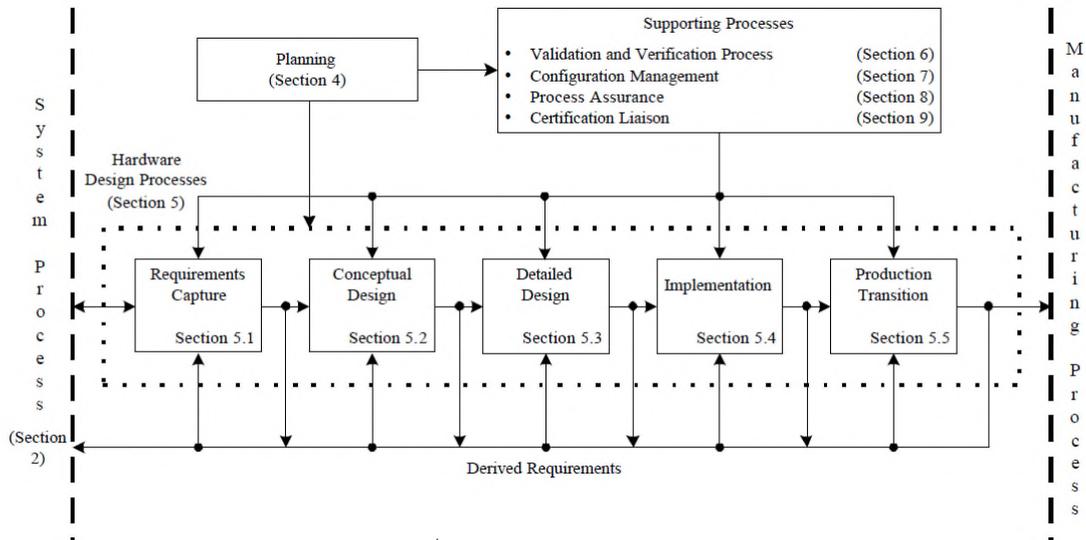


Figure 2. This is Figure 5-1 in the DO-254 document.<sup>1</sup>

A brief summary of the key points reflected in Figure 5-1:

- The **System Process** (DO-254 Section 2) provides the *design assurance level* (DAL) for the device and device requirements allocated from the system. *Derived requirements*, those created based on design implementation decisions, feedback to the safety analysis process performed at the system level.
- The **Planning Process** (DO-254 Section 4) includes information regarding the *Plan for Hardware Aspects of Certification* (PHAC), an important document developed during this part of the lifecycle that shows how all aspects of DO-254 compliance will be achieved. Once agreed to by the certification authority, this document guides the activities of the entire DO-254 project.
- **Requirements Capture** (DO-254 Section 5.1) involves having a mechanism to store and manage requirements. This is also tightly tied to requirements traceability which is covered under “Validation and Verification.”
- **Conceptual Design** (DO-254 Section 5.2) involves creating the architecture for the design that must implement the captured requirements. This is especially useful for complex designs. This stage may be merged with a detailed design if the design is relatively simple.
- **Detailed Design** (DO-254 Section 5.3) is the process that begins the actual development work and typically involves coding the design using a *hardware description language* (HDL). This phase involves the transformation of the code into a netlist during the synthesis process. The transition from Detailed Design to Implementation may be a grey area. The detailed design phase is when the “modeling” of the design is undertaken, prior to any hardware being available.
- **Implementation** (DO-254 Section 5.4) is the development phase after device modeling prior to manufacture (for ASICs) or program (for PLDs and FPGAs) the silicon.

1. Used with permission of the RTCA. To purchase a copy of DO-254, visit [www.rtca.org](http://www.rtca.org).

- **Production Transition** (DO-254 Section 5.5) occurs after the design work and the devices are ready to begin larger volume production.
- **Validation and Verification** (DO-254 Section 6) is one of the “Supporting Processes” of DO-254 that occurs throughout the hardware design. Validation ensures that the requirements are correct, complete, and verifiable. Verification ensures the device performs the intended function as specified by the requirements. Part of this process includes *traceability*. Requirements must be traced to their implementation (typically linkage is established to/from the HDL code) and verification (both test cases and results). Traceability of validated requirements helps ensure that all functions are implemented and working as expected.
- **Configuration Management** (DO-254 Section 7) helps ensure that the device is developed in a structured, repeatable, and controlled environment. This includes version management, problem reporting, and related processes to ensure that the development process can consistently replicate an item, regenerate pertinent information and make modifications to an item in a controlled fashion if necessary.
- **Process Assurance** (DO-254 Section 8) is an actual project role focused on ensuring that the processes defined in the PHAC are followed.
- **Certification Liaison** (DO-254 Section 9) involves engaging with a certification authority (typically a *designated engineering representative*, or DER, in the U.S.) to ensure DO-254 compliance during the development process. This typically involves four official audits, called “Stage of Involvement” or SOI audits, whereby DO-254 objectives are demonstrated to a certification authority and credit is given for meeting them. Likewise, compliance failures or “findings” are also noted and must be addressed before compliance is granted.

## MAPPING THE DO-254 LIFECYCLE TO A TYPICAL FPGA FLOW

Typical FPGA process flow diagram

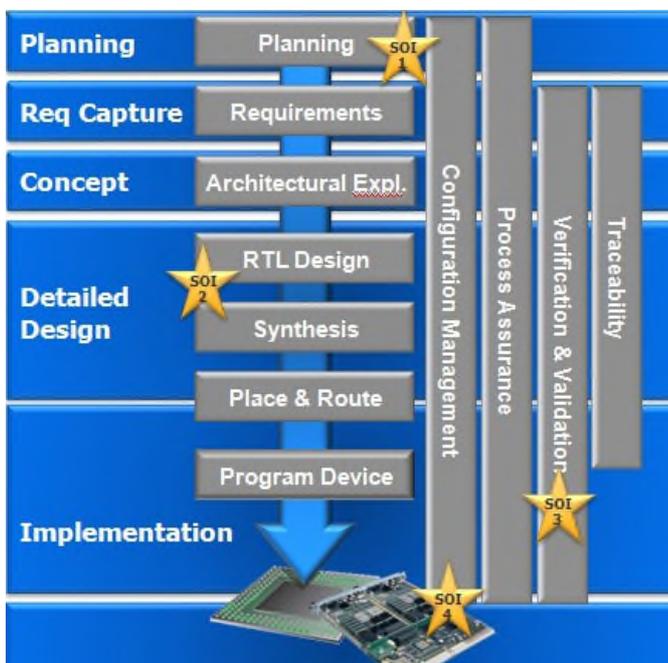


Figure 3. DO-254 lifecycle applied to an FPGA design process

More information about each of these processes and how they are done is presented in the **Solutions to Facilitate DO-254** section.

## UNDERSTANDING TOOLS AND TOOL ASSESSMENT

One key aspect of the DO-254 process is ensuring the tools used in the design process are working properly with respect to how they are being used within the target project. In DO-254 terms, this is called “tool assessment,” which may require a process called “tool qualification.” The purpose of tool assessment (and potentially qualification) is to ensure that the tools used to design and verify the hardware perform to an acceptable level of confidence on the target project.

“Tool Assessment and Qualification” is the topic of DO-254 section 11.4. It is important to understand tool assessment and qualification in terms of its usage in a DO-254 program, and how these processes differ from “Tool Qualification” in DO-178B so as to avoid potentially unnecessary work.

Figure 4 shows the flowchart Figure 11-1 from DO-254.

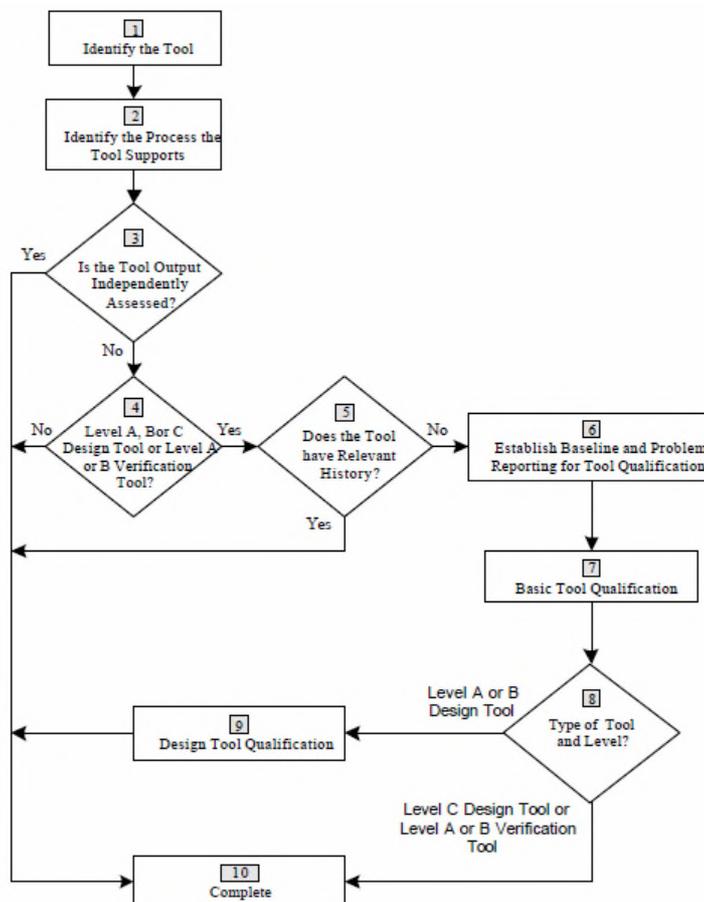


Figure 4. “Tool Assessment and Qualification” flowchart (Figure 11-1 from DO-254)

Tools are classified as either design tools or verification tools, depending on which design flow processes they automate. Likewise, designs are designated with a range of criticality levels (i.e., DAL) that correspond to the resulting severity of failure – from level A designs, which have the most catastrophic result if faulty, to level E designs, which have no safety impact. The rigor of the tool assessment and qualification process depends on both the tool classification as well as the DAL of the designated project.

Tool assessment and qualification can happen one of three ways:

1. **Independent Output Assessment**, where another independent tool or method validates the results of the tool.
2. **Relevant History**, where the tool has been previously used and shown to provide acceptable results.
3. **Tool Qualification**, which requires establishing and executing a plan to confirm that the tool produces correct outputs for its intended application on the particular project seeking compliance.

Regardless of these classifications, the task of tool assessment falls upon the airborne applicant or integrator who propose the method of tool assessment as part of the DO-254 planning and documentation. The certification agency or its representative (in the United States, this would be the DER) will determine if the proposed method of compliance is adequate for the development process. A tool vendor cannot assess or qualify their own tools and the FAA does not provide blanket approval for use of any tools in DO-254 projects.

For more information on this topic, refer to the companion whitepaper entitled “Assessing the ModelSim and Questa Tools for Use in DO-254 Projects” available at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).

## SOLUTIONS TO FACILITATE DO-254 COMPLIANCE

### PLANNING

The purpose of planning is to have a project team thoroughly think through aspects of the development process that will demand DO-254 compliance, document how compliance will be achieved, have this documentation reviewed and accepted by a certification authority, and then to have this information (in the form of a PHAC and the other planning documents) guide all the program activities. This process can break down in a number of ways, including when team members are not trained on DO-254 compliance. Training is more than purchasing RTCA/DO-254 and making it required reading. Many other factors, policy documents, and industry opinions have shaped the accepted and best-practice methods of compliance. The best way to learn all of this is to take DO-254 compliance training from a well-qualified trainer. (Mentor Graphics offers scheduled and on-demand, public, private and customized training classes on DO-254 compliance. For more information, visit [www.mentor.com/training\\_and\\_services/training/courses/fpga\\_pld/236322](http://www.mentor.com/training_and_services/training/courses/fpga_pld/236322).)

Another challenge of this phase is writing the plans themselves. Some companies provide templates that can help simplify the process. Another possibility is finding others within an organization that has already developed plans and leverages their work. Alternatively, simply look at the guidance in DO-254 section 10.1.1, which provides a clear outline of what needs to be included in a PHAC document. Supplement this by looking at section 3.1 “Activities for Stage of Involvement #1 – Planning Review” in the Job Aid, to prepare for the planning review audit.

---

### HARDWARE DESIGN PROCESSES

DO-254 hardware design processes encompass development activities from requirements capture through production transition:

#### REQUIREMENTS CAPTURE (INCLUDING MANAGEMENT AND TRACEABILITY)

Requirements define the intended function of a device, and as stated earlier (from the CFRs), a DO-254-compliant process ensures that a device performs its intended function. So the first step in requirements-driven flow is capturing the requirements. In addition to simply capturing requirements, system requirements allocated to a hardware item must be reviewed (validated), managed (to control changes and their impact), and traced to the pertinent design and verification activities. Likewise, *derived requirements*, or those derived from design decisions throughout the process, must go through these same processes and additionally have a feedback mechanism to the system safety engineers for validation at that level as well.

Many companies that serve the aerospace market use enterprise-level requirements management systems, such as the DOORS database product from IBM. DOORS provides a database mechanism to store and manage requirements and is capable of supporting large and complex systems. Other companies, such as subcontractors developing only a component in a larger system may use office productivity tools such as Microsoft Word or Excel to capture component-level requirements. Design and verification work must link back to these requirements, regardless of the format in which they are captured and stored. In DO-254, this linking is called requirements tracing or traceability.

Though capturing a static set of requirements is relatively easy, the same cannot be said of establishing a requirements-driven design flow and managing requirements as they evolve throughout a project. Mentor Graphics ReqTracer™ addresses this challenge by connecting requirements in their source (e.g., DOORS or Word) and linking them to design elements and verification artifacts. This closes the gap between the requirements capture environment and the actual development environment, as shown in Figure 5. The tool can also help validate requirements by facilitating requirements reviews, guiding verification activities based on requirements status, and providing push-button certification artifacts.



Figure 5. ReqTracer bridges the gap from requirements to development

ReqTracer integrates with Mentor Graphics native environments of HDL development, verification, and synthesis, and is flexible enough to adapt to nearly any other tool that would be used in a DO-254 development process. In addition to traceability and validation support, ReqTracer assists in project management by creating a visual depiction of project status, which shows the requirements that have and have not been designed and verified. ReqTracer can also generate the traceability matrices required to meet DO-254 traceability objectives.

In essence, ReqTracer provides a requirements-oriented project management environment from concept through implementation and supports the traceability needs of DO-254 projects. For more information, see the whitepaper “Using ReqTracer to Facilitate a Requirements-Driven DO-254 Compliant Design” available at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).

### CONCEPTUAL DESIGN

This is the phase in which a high-level design concept is developed from requirements. If a design is not very complex (a PLD perhaps) and being developed by a single designer, conceptual design can go very quickly and can be as simple as a block diagram describing the basic function and interfaces. Drawings in Microsoft PowerPoint or Visio are not uncommon artifacts of a simple conceptual design. However, even simple drawings should include safety-related or special test functions.

Conceptual design becomes more involved for complex devices worked on by teams of engineers. The team may spend some time exploring alternate architectures to determine the most appropriate one to perform the specified function. Mentor Graphics Vista Architect ([www.mentor.com/vista](http://www.mentor.com/vista)) can help facilitate this sort of tradeoff analysis.

## DETAILED DESIGN

Detailed design involves the bulk of the development work and spans hardware description language (HDL) coding through synthesis and also place and route, though some may argue that place and route falls under the implementation phase.

During the detailed design process, HDL code must be written to certain design standards, verified, reviewed internally, audited externally, kept under configuration management, and traced to program requirements. Mentor Graphics HDL Designer tool can help facilitate and automate many of the tasks that are often done manually in a DO-254 program.

- **Managing the design flow:** HDL Designer provides a structured work environment that among other benefits allows a team to define the tool versions and scripts that will be run throughout the process. Running a step in the flow is as easy as clicking the appropriate button, such as “Simulate” or “Synthesize.” HDL Designer supports Mentor Graphics design and verification tools and many FPGA vendor tools.
- **Creating/editing:** HDL design is a text-based method to describe physical hardware as well as its functional behavior. Since it is text based, it can be written via a simple text editor. However, this can be an intensive, manual process. HDL Designer provides an alternative: a suite of advanced design editors (as opposed to simple text editors) to facilitate design development, including Interface-Based Design (IBD) spreadsheets, a state-machine editor, an Emacs/vi-compatible HDL-aware text editor, and editors for block diagrams, functional truth tables, flow charts, and algorithmic state-machines.
- **Visualizing/documenting:** Often created manually, HDL documents and artifacts are important for improving the understanding of the design (important for design reviews and code reuse) and are especially useful if the design must be recreated in the future. HDL Designer automates the creation such documents and artifacts in the form of pictures or other design representations.
- **Code checking:** DO-254 requires that teams define the standards they are going to use in a design process, including the coding standards they must adhere to. These standards and rules help avoid downstream problems with the design or design process. In a DO-254 program, the design code can be manually reviewed against these standards as part of the design reviews; however, this can be a painful, time-consuming, expensive, and error-prone approach. A better method is to enlist the help of a tool to automatically do this sort of checking and then simply review the results as part of the design reviews. HDL Designer includes an HDL coding rules checker (or *linter*), that includes a set of predefined (and modifiable) rule sets. Noteworthy among these is the “DO-254 Ruleset,” a set of design checks derived from real project experiences with companies doing safety- and mission-critical design and compiled with input from approximately 20 members of the DO-254 User Group.
- **Reviewing/auditing:** The second DO-254 audit, or SOI-2, is typically a design review. Prior to the official certification design audit, the design team should have had numerous internal reviews that cover the architecture, changes to requirements, coding, tracing HDL code to the requirements, and so on. HDL Designer can help facilitate such reviews by specifying source code, diagrams, project hierarchy, and a variety of other pertinent data, and then capturing these in a HTML website, which enables project reviews across teams and geographies. The activities/results of the audit can also be captured into this HTML format to provide proof (an artifact) of the review process.
- **Managing versions:** Beyond the design data, teams need to manage the project throughout the design flow. HDL Designer helps in this regard, as well. The tool includes design (and configuration) management features that provide interfaces to common version management solutions, including Clearcase, Subversion SVN, Microsoft VSS, and Clisoft SoS. Or designers can take advantage of the CVS and RCS version management utilities included with HDL Designer.

- Tracing requirements:** As it's written HDL code should also be linked back to the appropriate requirements, which is a process called "tagging." The comprehensive editing environments of HDL Designer integrate with ReqTracer's "tagger" feature to facilitate the linking of an HDL implementation to its requirements source.

Figure 6 shows a number of the HDL Designer features explained here, including the graphical user interface, the flow buttons, the DesignChecker (for coding checking), and integration with version management tools.

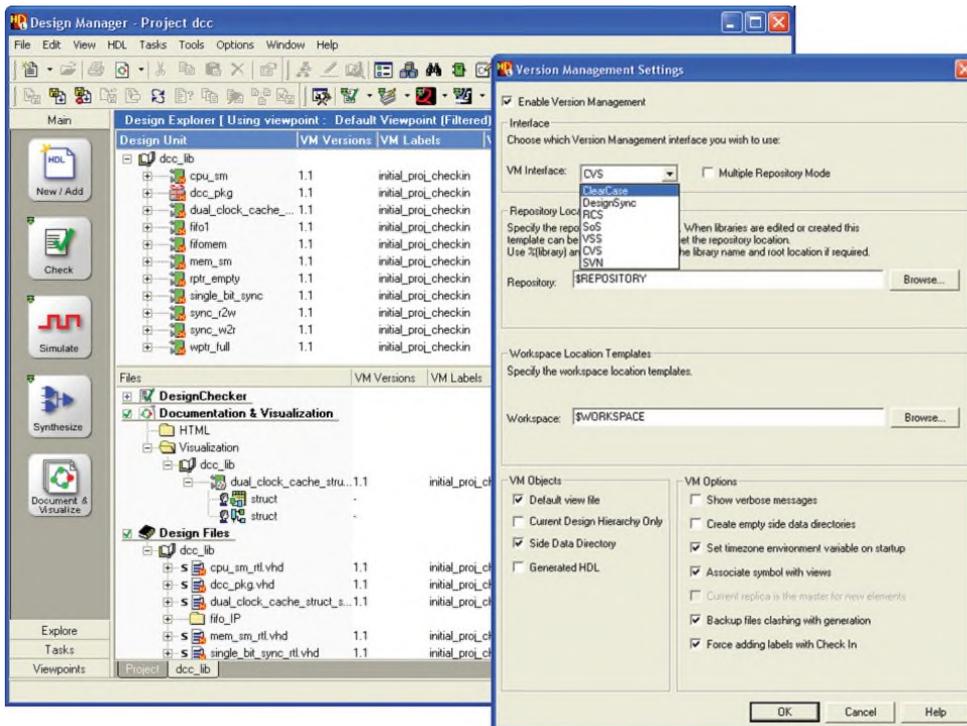


Figure 6. HDL Designer features

Synthesis, the heart of all modern PLD, FPGA, and ASIC design flows is a transformation of the HDL code into a technology-based netlist. Designers and their synthesis tools have tended to focus on achieving three main design goals: timing performance, design area, and tool run time. However, in military and aerospace applications where design assurance is critical a synthesis tool must take into account additional considerations.

Of course one such consideration is safety. Precision, a Mentor Graphics vendor-independent FPGA synthesis solution, balances aspects of safe synthesis with performance, optimization, and timing goals. The tool ensures that circuitry required for safe operation, such as specialized reset circuitry and special state machine encoding, is preserved during synthesis. The Precision Rad-Tolerant package addresses concerns about single event upsets (SEUs) from atmospheric radiation, by implementing triple modular redundancy (TMR) and safe state machine configurations during the synthesis process. It also supports the DO-254 principle of repeatability, providing a means to generate a deterministic and repeatable netlist given a consistent environment and conditions. In addition, it provides integration with the Mentor Graphics FormalPro™ logical equivalency checking tool to provide an added measure of assurance for the generated netlist. FPGA vendor software key to placement and routing of the netlist can also be launched directly from the Precision environment

For more information, refer to "Using HDL Designer to Facilitate DO-254 Compliant and Safety-Critical Design Processes," "Understanding and Running DO-254 Coding Checks in HDL Designer" and "Synthesis for DO-254 Design Assurance and Other Safety-Critical Design Processes" available at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).

## IMPLEMENTATION

Implementation is the move from detailed design to physical silicon package (e.g., FPGA). The back-end FPGA vendor tools support these tasks in a range of ways, from generating the bitstream to programming the device itself. The biggest implementation challenge is ensuring generation of a consistent, repeatable bitstream. Repeatability starts in the synthesis phase and must be carried through place and route and also through bitstream generation. Remember: one must be able to produce a repeatable bitstream prior to audit.

## PRODUCTION TRANSITION

Production transition is the phase for gathering information and processes necessary for consistently replicating the design on the manufacturing floor. Some people mistakenly think that manufacturing is beyond the scope of DO-254, but ensuring consistent replication is a DO-254 objective and many certification authorities will audit these processes in the production environment. Test out these processes in a clean environment prior to audit to ensure this objective is met.

---

## SUPPORTING PROCESSES

### VALIDATION AND VERIFICATION

System requirements allocated to the hardware item should be validated before design work begins. In addition, one must have a defined process for feeding the program's derived requirements back to the system and safety engineers for validation. Further, an established mechanism for identifying requirements attributes (e.g., validated, derived, safety-critical, etc.), as this can help track the appropriate activities associated with various categories of requirements, such as robustness testing of safety-critical properties and system level validation of derived requirements. ReqTracer, introduced in the section **Requirements Capture (Including Management and Traceability)** can assist with these validation tasks.

Verification is an ongoing process, and the many expectations associated with it are not entirely clear in the DO-254 document. What is clear is that for DAL A/B devices, verification must be an independent activity, which means the designer cannot test his own code. DAL A/B devices also must incorporate one or more "Advanced Verification" approaches from DO-254 Appendix B, the most common of which is "Elemental Analysis." Also, verification tests and results must trace back to the requirements.

What follows is a list of the typical (and usually expected) verification steps in a DO-254 program, though of course such steps vary depending on project and certification authority.

- **HDL simulation:** Because HDL simulation is generally expected, most companies develop complete sets of requirements-based tests. These are usually manually created "directed-tests" written by a verification engineer and corresponding to requirements, not to the design. A test set is complete when it covers all requirements. These tests should be reviewed, just as HDL code is reviewed, to ensure they are accurate and thorough tests. Keep proof that the review took place, something people commonly forget to do. The tests are then simulated, and the simulation results must trace to the requirements they support. The Mentor Graphics ModelSim® simulator is commonly used, as it is the dominant simulator in the military and aerospace industry.
- **Code coverage (in support of elemental analysis):** Elemental Analysis (from DO-254 Appendix B) is an advanced verification approach to ensure that the "elements" in a design are exercised by verification. In a typical DO-254 development program, where the design is developed in HDL, the "elements" become the elements in the HDL code (statements, conditions, finite state machines, etc.). Thus, running coverage analysis on the HDL code supplements simulation. Coverage gaps can reveal insufficient testing and unused code; in some cases such gaps may be the result of robust coding practices. Gaps must be closed or justified. ModelSim supports all pertinent metrics of code coverage. For more information, refer to the whitepaper "Code Coverage Explained for DO-254 Programs" found at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).

- **Clock-domain crossing analysis:** Integrating multiple functions into a chip is commonplace today and usually involves multiple, asynchronous clocks. Clock signals that cross domains can lead to a condition called metastability, which is a leading cause of device failure. The problems associated with signals that cross clock domains are extremely difficult and expensive to debug and fix because they typically are not detected until a failure occurs in the lab or field. Mentor Graphics 0-In@Clock Domain Crossing (CDC) is an analysis tool based on formal methods that can help reduce the likelihood of metastability. For more information, see “Mitigating the Dangers of Multi-Clock Designs” (presented at the 2008 FAA National Conference) found at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).
- **Static timing analysis:** Static timing analysis, STA, runs during synthesis and also place and route processes. STA analyzes paths to ensure they meet timing constraints. STA run during synthesis reports only estimated timing. STA run during place and route includes real delays as the design is being implemented into the real silicon. Unconstrained paths go unreported in STA, so it's important to properly constrain the design, and have this independently reviewed to ensure it's done properly. STA does not replace functional timing analysis (i.e., full timing simulation) and CDC analysis. The results of STA can be back-annotated onto the netlist for full timing simulation.
- **Netlist (full timing) simulation:** The transformations of synthesis and also place and route must be verified. This is typically done by re-running the HDL tests on the resulting netlist, with timing information from STA back-annotated. Because a netlist with timing information has much more detail than the mere functional description of the HDL code, this process can take quite a bit longer. Still, the general DO-254 expectation is that full test suite will be run. If the design is very large and complex, and the test suite takes days (or weeks) to simulate, other methods, such as logical equivalency checking may be considered. Mentor Graphics ModelSim tool can run this full-timing simulation as well
- **Hardware testing:** Verifying the device in its target system is the ultimate test as to whether the device performs its intended function. The device, after all functions as part of a system, and proper system function is what is critical for an aircraft. Prior to this phase all verification has tested the device either in isolation or within a model of the system. Testing the hardware (HW test) can occur either by 1) testing the silicon (programmed FPGA) in isolation or 2) in its target system. While the former is good practice, the latter is imperative. In support of No. 1, numerous new tools have entered the market to support integration of simulation and HW test. These tools reuse the simulation testbench on hardware, drive HW testing from the simulator, compare silicon to simulation results, and/or provide visibility into hardware for ease of debug. GateRocket ([www.gaterocket.com](http://www.gaterocket.com)) provides these sorts of tools, which can help ensure a very clean device to the final “in-target” (i.e., in-system) testing phase. However, they cannot take the place of the final system testing that DO-254 compliance requires. System level design and analysis can likewise support links to physical testing while also helping with the final system testing phase. The section **Looking Beyond the Chip** provides more information on this topic.

The following verification methods are less commonly used today but are nonetheless very powerful. Interest in and acceptance of these methods continues to grow.

- **Formal verification (model checking):** Listed as an acceptable method of advanced verification for level A/B devices in DO-254 appendix B, model checking is a formal methods technique that analyzes a design against its requirements, which are written as assertions. Model checking can exhaustively prove that a design performs its intended function. While not commonly used today on DO-254 programs, model checking is a very powerful verification technique for safety-critical design. 0-In Formal Verification is the Mentor Graphics model checking tool. For more information, see “Understanding Formal Methods for Use in DO-254 Programs” (presented at the 2009 FAA National Conference) available at [www.mentor.com/go/do-254](http://www.mentor.com/go/do-254).
- **Logical equivalency checking:** Logical equivalency checking, or LEC, is another formal methods technique that compares two models to ensure they are functionally equivalent. Distinct from model checking, LEC is commonly used in ASIC design flows and in fact can be used in many ways – usually comparing a golden model to a changed model. In a DO-254 program (even for FPGAs), it can be used to compare the input to

synthesis (or place and route) with the output to ensure the transformation was done correctly. LEC will very quickly (within minutes, not days) flag any functional differences. While LEC should not fully replace full timing simulation, it can be used as an added layer of assurance if running the full test suite is impractical. LEC does not address timing, so STA should always be run. Mentor Graphics FormalPro™ tool is a logic equivalency checker that can perform this task.

#### CONFIGURATION MANAGEMENT

Configuration management involves consistently replicating and regenerating an item, and modifying it in a controlled fashion if necessary. Version management tools such as Clearcase, Subversion SVN, Microsoft VSS, Clisoft SoS, CVS or RCS can significantly facilitate this process. Configuration management also includes problem reporting, which requires a structured process for tracking, managing and resolving problems found throughout the process.

#### CERTIFICATION LIAISON

Certification liaison stage includes three main stumbling blocks: not contemplating certification early enough, not including the certification liaison milestones in the PHAC, and not being prepared for SOI audits. To avoid these, begin communicating with a certification authority as early as possible – typically early in the planning process, which is the best time to discuss issues and concerns. The PHAC itself should include at least a tentative schedule of when the SOI audits are expected to occur (with an agreed mechanism as to how to handle schedule changes). Finally, in order to prepare for SOI audits, the Process Assurance resource (see below) should hold mock audits prior to the official ones.

#### PROCESS ASSURANCE

The person assigned the Process Assurance role monitors adherence to the PHAC, helping flush out internal problems before they pose certification risks and cause project cost increase. Thus, this role must be taken seriously.

## LOOKING BEYOND THE CHIP

While chips, or “custom micro-coded components,” are certainly important, ultimately it’s the system that must perform its intended function. These systems, of which the chips are a vital part, are also growing more complex, and design assurance is ever more important at these higher levels of design. Today, there is a big gap between the level of design assurance and compliance oversight applied at the chip and that applied to the system. In some cases, the airframe integrators themselves have been pushing beyond the current scoping of DO-254 and demanding compliance at higher levels of design – PCB or LRU level. At the same time, the system and safety documents, ARP 4754 and ARP 4761 are in the final stages of their “A” revisions. These documents are soon expected to become official regulatory policy just as DO-254 has become.

System design is by nature a process of decomposition that starts from specifications and results in a detailed, multi-discipline design. Indeed, systems can be thought of at three levels: 1) functional (where it’s easy to describe via specification what the system is supposed to do); 2) architecture (where the function of the system starts to be divided among the various disciplines); and 3) implementation (which eventually defines how the lower level functions are designed and built into whatever technology is most suitable).

Thankfully, technology is evolving to help with the design and analysis of systems at these stages, each of which is marked by increasing complexity. As an example, a functional description can now be created as an executable specification. So instead of a static document, one can actually run and analyze an executable version of the specification. This executable specification can then be the basis for architectural analysis and decisions about how to create the best architecture. When the proposed architecture is established, a platform can be created for the development and analysis of the system across multiple disciplines.

Mentor Graphics Bridgepoint™ supports high level system design activities of functional definition and architectural analysis. With Bridgepoint one can capture the system function in an executable model using the industry standard languages xtUML and/or SysML. This model, in turn can transform the system function into a nuanced consideration of architecture, including where to implement sub-functions and what other technologies are requirement for eventual implementation.

Once established, the architecture platform can be created in Mentor Graphics SystemVision™ tool, a unified environment that can be used for mechatronic system development at all levels of abstraction and across all disciplines (e.g., digital, analog, software, mechanical, etc.). SystemVision also links to LabVIEW, the test development tool from National Instruments.

Ultimately, DO-254 requires in-target (i.e., in-system) hardware test. SystemVision enables the pull in of HDL models (a drag and drop process in SystemVision) and simulation in the context of the system. One can develop a system-level, requirements-based tests in LabVIEW and exercise them on the SystemVision system model. Later, when the physical system is actually built, the same LabVIEW tests can be run on the physical system. Not only is this a very efficient flow, it also supports the testing of a “custom microcoded component” within its target environment, both in the virtually and in its physical instantiation, using the same test set.

Bridgepoint and SystemVision both integrate with ReqTracer for automating requirements management in the system development flow. For more information on tools that support system modeling and analysis, visit [www.mentor.com/products/sm](http://www.mentor.com/products/sm).

## SUMMARY AND CONCLUSION

RTCA/DO-254 is a document that establishes “Design Assurance Guidance for Airborne Electronic Hardware.” Regulatory policy mandates that PLD, FPGA, and ASIC devices for airborne applications must be designed in compliance with DO-254. Reading DO-254 itself is not enough to understand what compliance entails. This paper described the essential elements of the DO-254 document and what it means to comply. It also provided advice and described solutions for efficiently performing the tasks and meeting compliance throughout the DO-254 life cycle. In addition, it provided a sneak peek into system level design, where regulatory policy is imminent.

**For the latest product information, call us or visit: [www.mentor.com](http://www.mentor.com)**

© 2010 Mentor Graphics Corporation, all rights reserved. This document contains information that is proprietary to Mentor Graphics Corporation and may be duplicated in whole or in part by the original recipient for internal business purposes only, provided that this entire notice appears in all copies. In accepting this document, the recipient agrees to make every reasonable effort to prevent unauthorized use of this information. All trademarks mentioned in this document are trademarks of their respective owners.

<b>Corporate Headquarters</b> Mentor Graphics Corporation 8005 SW Boeckman Road Wilsonville, OR 97070-7777 Phone: 503.685.7000 Fax: 503.685.1204	<b>Silicon Valley</b> Mentor Graphics Corporation 1001 Ridder Park Drive San Jose, California 95131 USA Phone: 408.436.1500 Fax: 408.436.1501	<b>Europe</b> Mentor Graphics Deutschland GmbH Arnulfstrasse 201 80634 Munich Germany Phone: +49.89.57096.0 Fax: +49.89.57096.400	<b>Pacific Rim</b> Mentor Graphics (Taiwan) Room: 1001, 10F International Trade Building No. 333, Section 1, Keelung Road Taipei, Taiwan, ROC Phone: 886.2.87252000 Fax: 886.2.27576027	<b>Japan</b> Mentor Graphics Japan Co., Ltd. Gotenyama Garden 7-35, Kita-Shinagawa 4-chome Shinagawa-Ku, Tokyo 140-0001 Japan Phone: +81.3.5486.3033 Fax: +81.3.5486.3004
<b>Sales and Product Information</b> Phone: 800.547.3000	<b>North American Support Center</b> Phone: 800.547.4303			

