WP401 (v1.0.1) March 7, 2012

DO-254 for the FPGA Designer

By: Dagan White

The standard that governs the design of avionic components and systems, DO-254, is one of the most poorly understood but widely applicable standards in the avionic industry. While information on the general aspects of the standard is easy to obtain, the details of exactly how to implement the standard are sketchy. And once an entity develops a process that achieves compliance, the details of how compliance was achieved become part of the intellectual property of that entity. This white paper focuses on the details of developing a DO-254 compliant process for the design of FPGAs.

DO-254 - The General

DO-254, *Design Assurance Guidance for Airborne Electronic Hardware* [Ref 1], provides guidance for design assurance in airborne electronic hardware (AEH) to ensure safe operation. Rather than specify how to implement the standard or which test should be completed, it specifies the requirements for a process of design assurance and certification. It is the lack of specifics that causes uncertainty with the user community on how to develop a design assurance process that meets DO-254.

Per the standard, all flight hardware needs to be classified according to a design assurance level (DAL). The standard defines five levels regarding the safety and criticality of an avionic system (A to E). For example, engineers designing to level A or B face a much more rigorous test, verification, and documentation process than for levels C, D or E [Ref 2].

Central to DO-254 is the hardware life cycle, describing the general phases a project moves through, from initial planning to certification, including feedback loops to allow adaptation of requirements as necessary. Similar to other quality standards, DO-254 does not specify how to manage the life cycle nor the tools and methods to be used. However, it does require that design and certification procedures, methods, and tools be documented, along with the criteria used to determine when a project is allowed to move to the next phase.

Key to DO-254 is the designated engineering representative (DER) [Ref 3]. The DER is an appointed engineering resource who has the authority to pass judgment on aviation-related design and development, acting as the certification authority on behalf of the civilian aviation authorities. The standard allows the DER to be either an employee of the system developer or an independent consultant. Given that the DER must be approved by the civilian aviation authorities, the DER is often a consultant hired by the system developer. The DER has the authority to certify the process and can therefore assist in defining the process and the associated hardware life cycle.

A DER's involvement in a project depends upon the type and scope of a development project. Some example scenarios are:

- Self certify and submit without a DER: Very rare and only possible with system developers with extensive knowledge of DO-254 and aviation authority policy and a long history of safety.
- Internal DER: Common in larger avionics companies with sufficient activity to justify the expenses.
- Consultant DER: Common for line-replaceable-unit (LRU), board, or IP developers, or for companies whose main business is not avionics.
- DER as an auditor: A DER only audits the development on behalf of an aircraft integrator. This scenario can apply to board and IP developers where the end customer takes responsibility for managing the certification process.

DO-254 Hardware Life Cycle

DO-254 segregates the various activities of the hardware life cycle for complex electronic hardware (CEH) into one of three processes: planning, development, and correctness. In parallel to these three processes is the verification and validation process. Along with these three high-level processes, the standard also specifies what type of documentation must be kept (and delivered) to both manage and document the hardware life cycle. Again, DO-254 does not specify how each of these processes should be completed, but rather how each should be documented for review by the certification authorities.

Figure 1 shows the DO-254 hardware life cycle with the included hardware design process (section numbers shown refer to the applicable section of the standard). The various parts of the FPGA design process are shown, mapped to the hardware life cycle. In addition, processes that feed into and out of the hardware life cycle are shown.

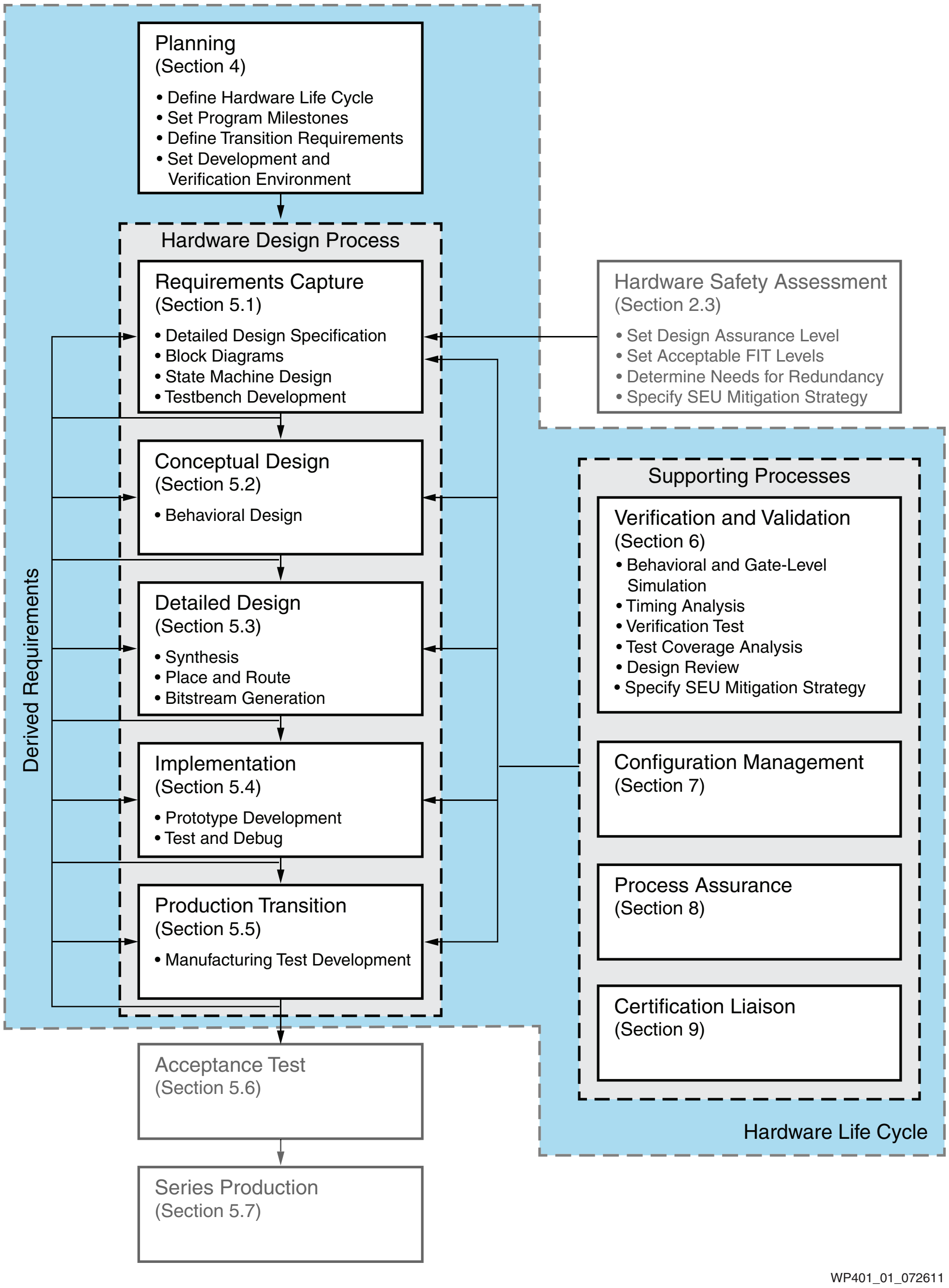


Figure 1: **DO-254 Hardware Life Cycle for Complex Electronic Hardware**

Planning

A plan for achieving DO-254 certification must be developed and documented for review by the DER. This plan, referred to as the plan for hardware aspects of certification (PHAC), summarizes the system's functionality, its architecture, the hardware and validation process, and for levels A to C, verifies fail-safe operation of the system using different redundancy strategies. The PHAC also must include details regarding the development environment, system testing, the different phases of system hardware development, and the transition criterion. In addition, the mechanism for incorporating derived requirements (in other words, detailed requirements stemming from work in later phases) must be covered.

Along with the PHAC, a top-level drawing identifying all assemblies, subassemblies, and components must be delivered. For FPGA development, this includes a block diagram showing each functional partition and the I/O between functions, including a description of each.

After the PHAC is approved by the DER, development, testing, and implementation of the system can begin.

Hardware Safety Assessment

While not considered a part of the hardware life cycle by DO-254, the hardware safety assessment does directly impact FPGA design. This assessment determines the DAL for each functional block in the system. A system developer has the option of setting a single design assurance level and strategy for an entire hardware item, or a hardware item can be determined to have separate functional failure paths (FFPs)—and as a result, a separate DAL and assurance strategy.

FPGAs, similar to SRAMs, possess a unique FFP—susceptibility to single-event upsets (SEUs). When high-energy particles (primarily neutrons) pass through the silicon substrate of a device, charged particles are created as the result of collisions. If the charge of these particles is sufficient, they can change the state of a static memory element. These changes of state are referred to as SEUs. With an SRAM FPGA, not only do SEUs occur in the device memory but in the configuration memory as well, potentially creating a change in the FPGA's programmed logic.

Because of the increased neutron flux at high altitude (the flux at 40,000 feet can be roughly 600 times that of a ground-based observe), the potential of configuration upsets must be included in any assessment. Moreover, mitigation strategy must be selected based upon the predicted FIT[1] rate as well as the required DAL. For more details on the impact of SEUs in avionics and various mitigation techniques, see the Xilinx Avionics Developers' website [Ref 4].

The output of this stage is a system safety assessment (SSA) detailing the DAL for the item, the assurance strategy to be pursued, and the recommended SEU mitigation strategy.

[1] To assist customers in predicting FIT rates prior to mitigation for selected Xilinx® FPGAs, Xilinx has developed an SEU FIT rate calculator. Requests for the FIT rate calculator can be sent to avionics@xilinx.com.

Hardware Design Process

The hardware design process phase is broken into five distinct sub-processes that must be documented:

- Requirements capture: The architecture of the system (and the system-level requirements), including items such as test structures and interfaces, needs to be described and documented. During this phase, the design team must develop a block-level description of the system, including block diagrams, state diagrams, and flow charts, that are consistent with the requirements.
- Conceptual design: During this phase, hardware design can begin with HDL development. The output of these activities plus the results of preliminary design review are submitted to the DER for review.
    - Simulation, while part of verification and validation, is considered a natural part of the conceptual and detailed design processes.
- Detailed design: During this phase, the design is synthesized, and place-and-route is completed. After the confidence in the design is high, bitstreams are generated.
- Implementation: During this phase, FPGAs are programmed and prototypes are developed to allow test and debug.
- Production transition: In the last phase, the FPGA/board is prepared for release to manufacturing. After the board and FPGA are fully debugged, test engineering completes production and reliability testing, a baseline is established to ensure consistent system production, acceptance testing is defined.

Supporting Processes

Running concurrently with the development processes, the supporting processes are designed to manage certification activities and ensure that the top-level requirements have been implemented in hardware as intended.This collection consists of four major processes:

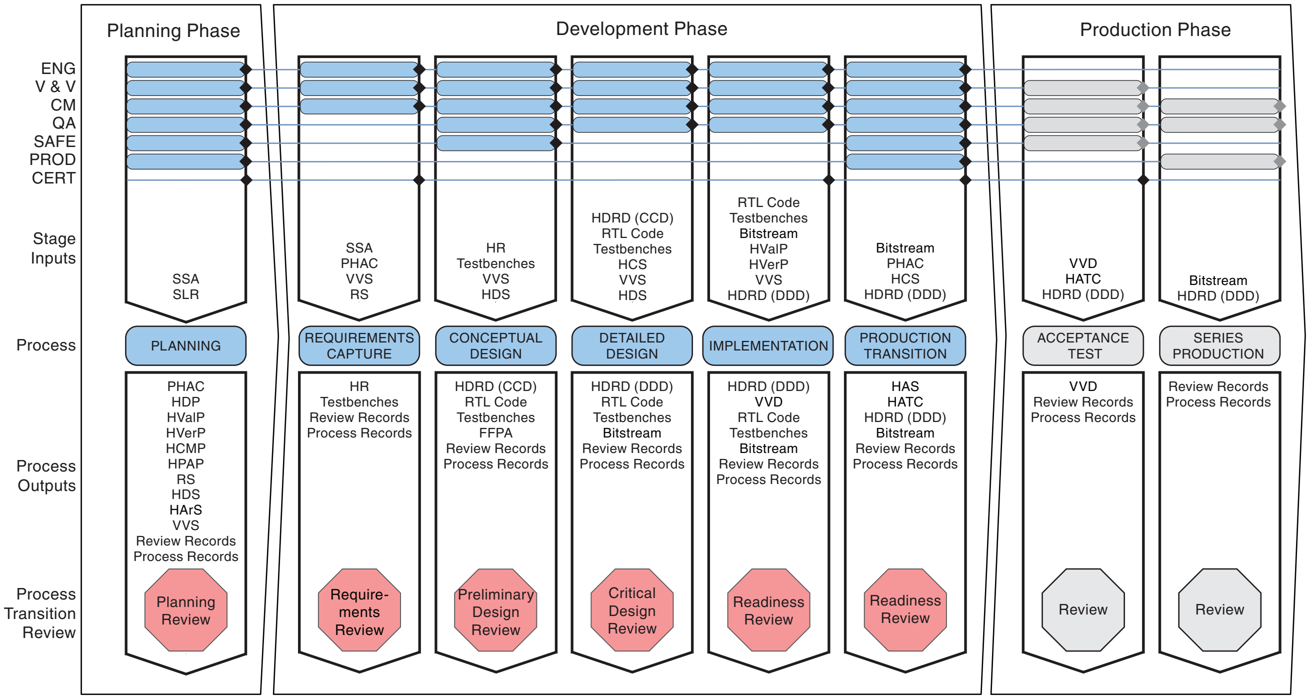
- Validation and verification: Validation assures that derived requirements (especially those relating to safety) are correct and complete with respect to system requirements. Verification assures that the hardware implementation meets all of the hardware requirements, including derived requirements. Verification activities typically involve simulation at all levels (behavioral, structural, back-annotated timing), static timing analysis, fault simulation, test coverage analysis, and design reviews.
- Configuration management: This process archives all data needed for certification and all data needed to return production to any earlier version.
- Process assurance: This activity ensures that hardware life cycle data and processes comply with the planning documents.
- Certification liaison: This process defines the communication between the system developer and the certification authority (usually, the DER), covering how data is approved, joint reviews, and when and how the certifying authority witnesses compliance testing.

Documentation and Organization

No standard is complete without a requirement for documentation. The DO-254 standard refers to artifacts, which are documents, reports, results, standards, or design files that are developed or generated as part of the development process. While the standard does not specifically set what data items are needed, it does provide general guidelines for the type of documents, reports, and data needed and in what context. Ultimately, the DER reviews the artifacts as part of the certification process. Having the artifacts properly controlled and linked streamlines this process.

The development organizational structure is the responsibility of the system developer. While it might be inferred that the standard requires parts of the design process be handled by different organizations, that is not the case.

Figure 2 provides a detailed breakdown of the type of documents needed as inputs and outputs for each phase of the FPGA design process, plus an indication of what organization function is involved in a particular phase. The production phase is shown for completeness but not described in this white paper.



ENG: Cross Functional engineering team. Includes systems engineering and detailed engineering teams
V&V: Test engineers who perform verification and validation tasks, separate from the engineering design teams
CM: Configuration control and management personnel
QA: Quality control and process assurance personnel
SAFE: Safety analysis and assurance personnel
PROD: Production team, including procurement, assembly, and test.
CERT: Certification liaison, DER or equivalent who represents or interacts with the certification authority (FAA, EASA, etc.).

WP401_02_080311

Figure 2: **Detailed Breakout of the DO-254 Design Process for FPGAs**

DO-254 and the FAA

The FAA issued order 8110.105 [Ref 5] in 2008 to clarify the how and when certification staff (including the DER and FAA employees) should apply DO-254. The scope of the order is restricted to simple and complex hardware components (i.e., applies only at the device level).

The order impacts the hardware life cycle with regards to:

- Design reviews
- Defining the level of FAA involvement
- Additional topics regarding DO-254 application

FAA-Mandated Reviews

DO-254 states that reviews are needed at the end of each stage to assess design data and readiness to proceed to the next stage. Order 8110.105 [Ref 5] further defines these reviews, referred to as stages of involvement (SOI), at specific project milestones. While the order describes four review milestones, it does allow for some of these reviews to be combined at the discretion of the certification authorities, based upon the DAL and complexity of the system.

The four reviews or SOIs defined by the order are:

1. Hardware planning review, presumably at the end of the planning process.
2. Hardware design review, typically conducted sometime between the midpoint and completion of hardware design data (requirements, design, and implementation).
3. Hardware validation and verification review, again sometime after the halfway mark of hardware validation and verification process.
4. Final hardware build and verification review. A hardware conformity review is done when the system is ready for formal approval.

These reviews are recommendations. The exact timing and goals of each review is negotiated with DER during the planning phase.

Level of FAA Involvement

FAA order 8110.105 [Ref 5] describes a scoring process by which the FAA certification authority determines how much direct FAA involvement must occur in a project. The order defines three broad levels of involvement:

- High: Minimal delegation to the DER. The DER can recommend data approval, but the FAA chief scientific and technical advisor (CSTA), directorate staff, or headquarters staff must be involved throughout the hardware life cycle.
- Medium: The DER can recommend approval of the PHAC, HAS, and possibly the HCI. FAA involvement during the planning, regulation, and policy interpretation as well as final approval. There must be at least one onsite review at this level.
- Low: The DER can recommend approval of the PHAC but can approve all other plans and data. There is little to no involvement with the FAA.

Additional Topics in DO-254

FAA order 8110.105 discusses a number of additional topics, giving guidance on when and how to apply DO-254. One of these topics describes different applications for a PHAC. The order provides three different scenarios:

- A PHAC can be written for each electronic hardware component to support reuse in multiple systems.
- All electronic hardware components of a system can be combined into a single PHAC.
- A combined PHAC, including other planning data for the aircraft or system, can be created to form a project-specific certification plan (PSCP).

FPGA Tool Flow for DO-254 Projects

A key to success with DO-254 certification is having a well defined and structured hardware design process. As defined in Figure 1, there are five distinct sub-processes of the hardware development phase plus the common processes that support the entire hardware life cycle. Two types of tools are used during the DO-254 hardware life cycle: development tools and supporting process tools. All tools must be specified in the PHAC and must be managed based on the DAL.

FPGA development tools typically consist of design capture, synthesis, place and route, and bitstream generation. These tools reside in the direct path from design entry to bitstream generation. Without any one of these tools, the user cannot build the final FPGA bitstream. These tools are subject to assessment and qualification under the DO-254 standard because they generate a hardware item or design.

The supporting process tools are commonly used across the hardware design process and include those used for simulation and verification, requirements management, and general documentation. These tools do not require assessment or qualification, except in the case of simulation and verification tools that are used to confirm that requirements have been met.

The requirement for tool assessment/qualification has generated much confusion in the industry as to the intent of the standard. First, this process is only required for design tools when used on hardware determined to be DAL A to C, and for verification tools that are used on hardware determined to be DAL A or B. When used on all other levels, no assessment/qualification is required.

For those instances where assessment/qualification is required, DO-254 provides two methods for completing this process: assessment or qualification (Figure 3).

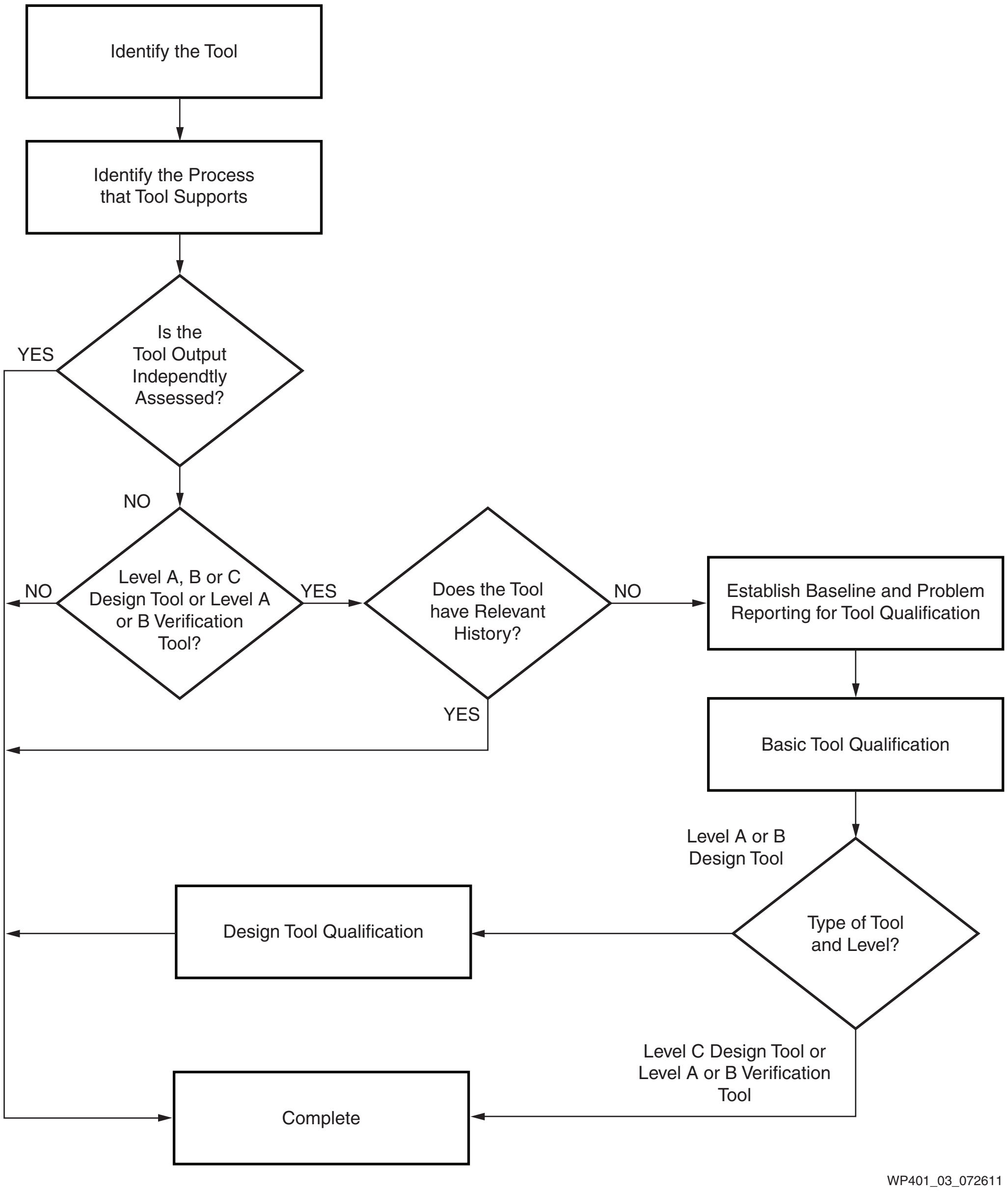

Identify the Tool
Identify the Process that Tool Supports
Is the Tool Output Independtly Assessed?
YES
NO
Level A, B or C Design Tool or Level A or B Verification Tool?
NO
YES
Does the Tool have Relevant History?
NO
YES
Establish Baseline and Problem Reporting for Tool Qualification
Basic Tool Qualification
Level A or B Design Tool
Type of Tool and Level?
Design Tool Qualification
Level C Design Tool or Level A or B Verification Tool
Complete
WP401_03_072611

Figure 3: **DO-254 Tool Assessment and Qualification Flow**

Assessment

If the output of the tool can be independently assessed (verified for correctness) by either manual review or automatic means, the tool has passed this process (no further assessment is required). For example, the output of a synthesis tool can be verified by comparing simulation results between the input HDL and resulting netlist, similarly, the output of a place and route tool can be verified via simulation.

If (and only if) the tool output is not independently assessed, then the service history of the tool can be used to demonstrate that the tool has been properly assessed. Almost without exception, all commercial design tools meet this last requirement.

For example, the Xilinx ISE® suite of design tools has had a long history of use, not only for design capture but also for place and route. For more than twenty years, all Xilinx FPGA designs have been placed and routed with ISE software. This history of usage on hundreds of thousands of designs world-wide constitutes a continuous and independent assessment of its performance, providing relevant history for use on DO-254 projects.

Qualification

If a tool does not meet the requirements of assessment, then it must be qualified. The first step in qualification is establishing configuration management and problem reporting for the tool. The next step is to create (and execute on) a qualification plan (possibly along the lines of RTCA DO-178B/EUROCAE ED-12B used for software development tools). Qualification of a tool not previously used can be a lengthy process and should be discussed with the DER during the planning phase of a project.

Tool assessment and qualification is only valid for a specific version of the tool, and any new versions must be reassessed to be qualified for use in DO-254 projects.

Tool Flows for FPGA Design

Development and support tools must be considered for each subprocess of the DO-254 Development Life Cycle. A supporting tool flow demo can be found on the Xilinx Avionics Developers' website [Ref 4].

Requirements Capture

The requirements capture subprocess begins with the review of the system safety assessment (SSA), PHAC, validation and verification standards (VVS), and the requirements standards (RS) documents from which the hardware design process starts. The SSA, PHAC, and VVS documents are developed during the planning phase are used in the requirements capture process by the cross functional engineering team (ENG), the test and verification engineering team (V&V), and the configuration management team (CM).

This phase of the design process defines system architecture considerations, functionality, and performance. The requirements are typically captured in a requirements database tool, which organizes the requirements in a hierarchical methodology. The parent-child requirement relationships can be associated to demonstrate satisfaction of the parent requirement through the satisfaction of the child requirements. The requirements database tool can be targeted for this specific activity—or can be as common as a standard spreadsheet or word processing application.

At this point, the engineering and the validation and verification teams need to set standard naming conventions for both design and verification requirements for the purposes of automating requirements tracking and reporting. Tools are available to automate the requirements tracking and reporting from specification to implementation and verification through various documents types and databases sources throughout the development project. Key to enabling these types of tools is the use of standard naming conventions for requirements capture and design implementation. The focus of DO-254 is definition, design, and verification of requirements. Automating this process greatly accelerates review check points and it aids in demonstrating compliance.

After the hardware requirements (HR), verification requirements, and testbenches are specified, the next step is to develop the derived requirements and begin the conceptual design processes.

Conceptual Design

The conceptual design phase utilizes the HR, testbenches, VVS, and the HDS in the development of functional block diagrams and state machines plus the associated interfaces that implement the hardware requirements. In parallel to this development task, functional failure path analysis (FFPA) must be performed to identify failure paths of the design and to analyze how the design architecture can address or mitigate the failure path to meet the required DAL.

Tools such as Xilinx ISE design suite can be used to develop high-level block diagrams and state machines and generate the HDL code at the structural level, with the lower level design to be completed in the detailed design phase. Functional failure paths (FFP) must be captured, documented, linked, and traced to the hardware requirements and derived requirements. The verification process at this level begins with the design of the verification methodology required to meet the DAL. Techniques ranging from simple directed testing to assertion-based verification can be implemented with commercially available verification tools [Ref 4].

As the design is developed to the given requirements, and derived requirements are generated, it is important to maintain the consistency of the naming convention for these requirements. A design review is held with the DER to ensure that the output of the conceptual design meets the hardware requirements.

At this review, the following documentation or representation must be available for review by the DER: hardware design representation data (HDRD) consisting of the conceptual design data (CDD) typically represented as high-level block descriptions and/or block diagrams, RTL code at the interface levels, testbenches at the behavioral level, and FFPAs.

Detailed Design

The detailed design process takes the outputs of the conceptual design process along with the hardware coding standards (HCS), validation and verification standards (VVS), and the hardware design standards (HDS) as defined by the PHAC and begins the detailed development of the RTL code and testbenches. The tools in the conceptual design phase, such as ISE design suite, are used to perform the detailed implementation of the RTL code. Additional tools, for example, linting tools, can be used to implement or verify coding standards.

As the RTL code is implemented in detail and verified accordingly, the next step is to synthesize the design with a tool such as ISE XST software. The resulting gate-level netlist is used to verify whether the design requirements have been met. This netlist is subsequently used as input to the ISE place and route software to generate back-annotated SDF files for timing verification as well as to generate the programming bitstream for the target FPGA.

Verification at the detailed design stage must be made to ensure that the safety and design requirements are satisfied. At this stage, verification methodologies such as directed testing and constrained-random approaches, should be considered based on the complexity of the design. Formal equivalence testing of the gate-level netlist against the RTL code can be used to employ mathematical methods to exhaustively verify behavior and functionality of the design, which would be difficult or nearly impossible to prove using directed testing or constrained-random techniques. Additionally as part of verification, code coverage from a functional, statement, conditional, and branch perspective must be verified to ensure that the RTL code that implements the design is used as part of the design.

The output of the detailed design phase, the HDRD with the detailed design data (DDD), the RTL code, testbenches and programming bitstream of the FPGA, are used as inputs to the implementation process.

Implementation

For programmable logic, the implementation phase starts with taking the bitstream generated during the detailed design phase and programming the target FPGA. Initial testing and verification (design debug) can be completed on test beds, but the implementation phase cannot end until the design has been verified in a production-ready system. Standard design debug tools apply here: signal generators, logic analyzers, device programmers, etc.

The design can progress to the product transition phase after:

- The implementation of all requirements is verified.
- The design flow generates the same bitstream repeatedly using the final design data.

Any errors and omissions in the requirements must be fed back to the requirements phase. Derived requirements and design errors are fed back to the detailed design phase.

Product Transition

The goal of production transition is to ensure consistent and repeatable manufacture of the target system. Implicit in this process is the development of production test programs. Simulation tools apply in this phase to verify fault coverage of planned tests.

The product transition phase is not complete until:

- All data (procedures, test programs, programming files, etc.) required for repeatable system manufacture is identified and developed.
- Any manufacturing steps that can impact safety are identified and documented.

Conclusion

DO-254 was developed to address issues of design assurance and certification challenges for avionics using complex electronic hardware. Because the standard provides only guidance rather than specify exact processes and methods, it is written broadly, resulting in confusion. Given the high level of this guidance, this confusion extends to understanding how DO-254 applies to FPGA design. While this white paper attempts to add clarity, ultimately, each organization has to determine how to apply this guidance to their design process in consultation with the appropriate certification authorities.

Xilinx understands the commercial avionics market needs in relation to DO-254 and is working with the industry to improve the application of DO-254 to FPGAs. For more information, go to the Xilinx Avionics website.

References

1. DO-254, *Design Assurance Guidance for Airborne Electronic Hardware*, published by RTCA, Inc.
   http://www.rtca.org/onlinecart/product.cfm?id=194
2. WP332, *Meeting DO-254 and ED-80 Guidelines When Using Xilinx FPGAs*
3. AIR-140 CEH Memorandum, *Designated Engineering Representative (DER) Authority for Complex Electronic Hardware Approval and Special Delegation for TSO Complex Electronic Hardware*
4. Xilinx Avionics Developers' website: www.xilinx.com/member/avionics
5. FAA Order 8110.105 CHG 1, *Simple And Complex Electronic Hardware Approval Guidance*

Process Documents Referenced in DO-254

- ATP: Acceptance Test Plan
- CDD: Conceptual Design Data
- DDD: Detailed Design Data
- EAR: Elemental Analysis Results
- FFPA: Functional Failure Path Analysis
- HATC: Hardware Acceptance Test Criteria
- HArS: Hardware Archive Standards
- HAS: Hardware Accomplishment Summary
- HCMP: Hardware Configuration Management Plan
- HCS: Hardware Coding Standards
- HDP: Hardware Design Plan
- HDRD: Hardware Design Representation Data
- HDS: Hardware Design Standards
- HECI: Hardware Environment Configuration Index (combine with HCI)
- HPAP: Hardware Process Assurance Plan (QA Plan)
- HR: Hardware Requirements
- HValP: Hardware Validation Plan
- HVerP: Hardware Verification Plan
- HVR: Hardware Verification Results (procedures and results in one file)
- PHAC: Plan for Hardware Aspects of Certification (PHAC is largely generic across IPs)
- RS: Requirements Standards
- SLR: System Level Requirements (not a DO-254 artifact)
- SSA: System Safety Assessment
- SVR: System Verification Results
- VVD: Validation and Verification Data
- VVS: Validation & Verification Standards (include in HVVP)

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
09/07/11	1.0	Initial Xilinx release.
03/07/12	1.0.1	Minor typographical updates.

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.