

**Hardware Verification & Validation Plan  
  
For The  
  
Avionics Passenger Counter**

Document No: 800-HVVP-01

Revision: 0.1

|  |  |  |  |
| --- | --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | | \_\_\_\_\_\_\_\_\_\_\_ |
| Project Manager |  | Date | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_ | |
| Technical Project Lead |  | Date | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_ | |
| Principal Design Engineer |  | Date | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_ | |
| Principal Quality Engineer |  | Date | |

|  |
| --- |
| **Notice**  This document and the information contained herein are the property of Airworthiness Certification Services LLC. Any reproduction, disclosure or use thereof is prohibited except as authorized in writing by Airworthiness Certification Services LLC. Recipient accepts the responsibility for maintaining the confidentiality of the contents of this document. |

|  |  |  |  |
| --- | --- | --- | --- |
| REVISIONS | | | |
| Rev | Reason/Description | Requested/  Changed By | Date |
| 0.1 | Initial draft | N/A |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**Section Page**

1.0 INTRODUCTION 5

1.1 Purpose 5

1.2 Scope 6

1.3 Acronyms and Abbreviations 6

1.4 Applicable Documents 7

1.4.1 External Documents 7

1.4.2 Internal Documents 7

2.0 VALIDATION AND VERIFICATION METHODS 8

2.1 Review Methods 12

2.1.1 Review Verification and validation plans 14

2.1.2 Determine Need for Tool Qualification 15

2.1.3 Review Data Related To Qualification Tools 16

2.1.4 Review Test Cases and Procedures 16

2.1.5 Review Checklists for Test Cases, Procedures, and Results 17

2.1.6 Effectiveness of the Test Program 18

2.1.6.1 Assess Results of Requirements-Based Tests 18

2.1.6.2 Assess Failure Explanations and Rework 18

2.1.6.3 Assess Elemental Analysis coverage achievement 19

2.2 Analysis Of Outputs Methods 20

2.2.1 System And Hardware Requirements Trace Analysis Method 20

2.2.2 Hardware Requirements Specification Trace Analysis Method 20

2.2.3 Requirements-Based Test Coverage Analysis Method 21

2.2.4 Hardware Reviews and Analysis 21

2.2.4.1 Reviews and Analysis of High-Level Requirements 21

2.2.4.2 Reviews and Analysis of Low-Level Requirements 22

2.2.4.3 Reviews and Analysis of Hardware Architecture 23

2.2.4.4 Reviews and Analysis of Outputs of the Integration Process 23

2.2.4.5 Reviews and Analysis of Test Cases, Test Procedures and Results 23

2.3 Coverage Methods 24

2.3.1 Requirements Coverage 24

2.3.2 Simulation 25

2.3.3 Static Timing Analysis 25

2.3.4 Post-Route Simulation 25

2.4.1 Test Environment 28

2.4.1.1 Requirements-Based Test Cases 28

2.4.1.2 Normal Range Test Cases 28

2.4.1.3 Robustness Test Cases 28

2.4.1.4 Requirements-Based System Verification Testing Methods 28

3.0 Verification Data 29

4.0 Verification Independence 30

5.0 Verification Environment 31

5.1 Tools 31

5.2 Functional Tests 31

5.3 HDL Simulation 31

5.4 In-Circuit Test 31

5.5 Analysis Test 32

Appendix A: Requirements Review Checklist 33

Appendix B: Design Review Checklist 35

Appendix C: Verification Cases, Procedures, and Results Review Checklist 37

Appendix D: Production Transition Review Checklist 39

Appendix E: Functional Test Standard 41

Appendix F: HDL Simulation Test Standard 43

Appendix G: In-Circuit Test Standard 45

Appendix H: Analysis Standard 47

Appendix I: Implementation Review Checklist 49

# 

# 1.0 INTRODUCTION

## 1.1 Purpose

This plan describes the procedures, methods, and standards to be applied and the processes and activities to be conducted for the validation of the hardware item-derived requirements to achieve the validation objectives of this document. This plan also describes the procedures, methods, and standards to be applied and the processes and activities to be conducted for the verification of the hardware items to achieve the verification objectives of this document.

The verification process ensures that the product is built as designed, with no unexpected functionality. The verification process is also intended to ensure that the hardware will perform under any foreseeable operating conditions.

## 1.2 Scope

This plan will be used by the certification authority to determine if the Hardware Life Cycle Process is commensurate with the rigor required for the level of Hardware being developed. Once approved, it is implemented during the development and product life cycle of the deliverable airborne Hardware. This Hardware Verification & Validation Plan complies with the documentation requirements of RTCA/DO-254.

## 1.3 Acronyms and Abbreviations

AIMS Action Item Management System

ALU Arithmetic Logic Unit

ARP Aerospace Recommended Practice

ASIC Application Specific Integrated Circuit

DRMS Document Review Management System

HC1 Hardware Control Category 1

HC2 Hardware Control Category 2

COTS Commercial-Off-The-Shelf

EUROCAE European Organization for Civil Aviation Equipment

FAR Federal Aviation Regulations

FFP Functional Failure Path

FFPA Functional Failure Path Analysis

FHA Functional Hazard Assessment

F-FMEA Functional Failure Modes and Effects Analysis

FTA Fault Tree Analysis

HDL Hardware Description Language

HRD Hardware Requirements Document

HVCP Hardware Verification Cases and Procedures

HVVP Hardware Verification & Validation Plan

JAR Joint Aviation Requirements

LRU Line Replaceable Unit

PHAC Plan for Hardware Aspects of Certification

PLD Programmable Logic Device

PSSA Preliminary System Safety Assessment

RTMS Requirements Traceability Management System

SAE Society of Automotive Engineers

SC Special Committee

SSA System Safety Assessment

WG Working Group

HPA Hardware Quality Engineer

SRS Hardware Requirements Standard

SSA System Safety Assessment

VR Verification Results

VSS Visual Source Safe

## 1.4 Applicable Documents

The following documents are listed for reference only. Each document is applicable to this plan only to the extent specified herein.

### 1.4.1 External Documents

RTCA/DO-254 Design Assurance for Airborne Electronic Hardware

FAA Order 8110.4C Type Certification

AC 20-152 Advisory Circular, RTCA Inc., Document DO-254, Design Assurance for Airborne Electronic Hardware

### 1.4.2 Internal Documents

800-PSAC-01 Plan for Software Aspects of Certification

800-HVVP-01 Hardware Verification & Validation Plan

800-HCMP-01 Hardware Configuration Management Plan

800-HPAP-01 Hardware Process Assurance Plan

800-HRD-01 Hardware Requirements Document

800-HRS-01 Hardware Requirements Standard

800-HDS-01 Hardware Design Standard

800-HCI-01 Hardware Configuration Index, Passenger Counter FPGA

880-HCI-01 Hardware Configuration Index, Arinc 429 I/O FPGA

800-HTP-01 Hardware Review, Analysis & Test Procedures

800-HTR-01 Hardware Review, Analysis & Test Results

800-HAS-01 Hardware Accomplishment Summary

# VALIDATION AND VERIFICATION METHODS

This section includes a description of and references to the validation procedures, standards, and methods to be used on the Avionics Passenger Counter project. Figure 2-1 depicts the complete FPGA design flow which shows the verification and validation life cycle steps. Methods include analyses, reviews, and testing. Methods include analyses, reviews, and testing. This section also includes a description of the methods for the applicable verification completion criteria.

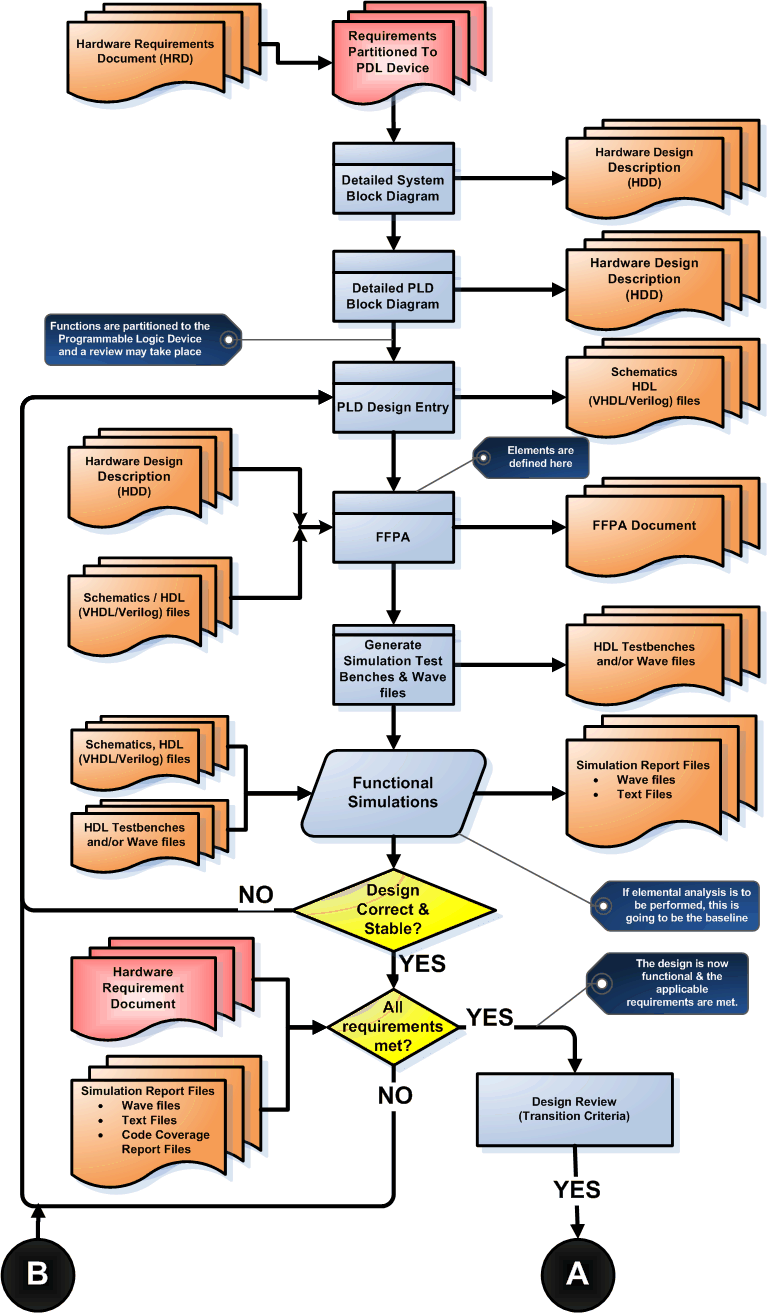
As subsets of the Requirements Data or Design Document are reviewed, the derived requirements in those portions shall be validated. From the system level requirements, both high-level and low-level requirements may be derived. The high-level requirements will address what needs to be done to satisfy a system level requirement. The low-level requirements (when necessary) will address how the high-level requirements have been satisfied.

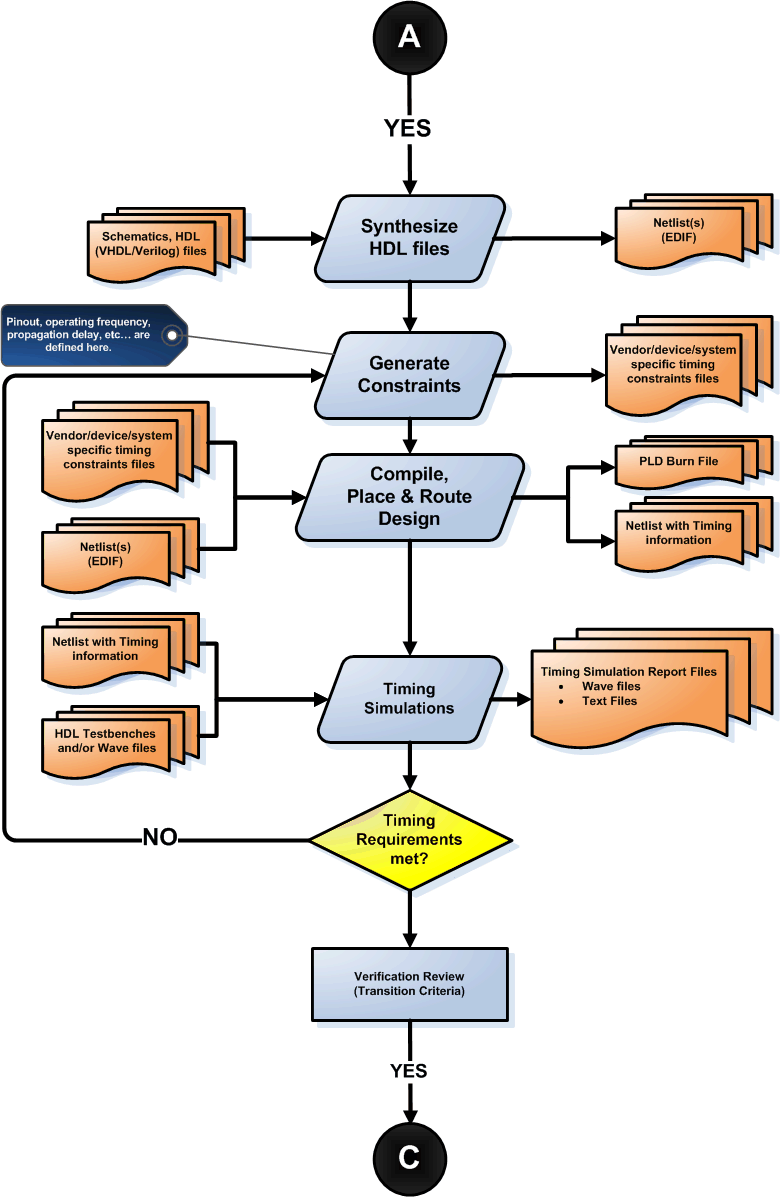
In order for validation to be complete, both a review of the requirements to the requirement standards and an analysis of the traceability shall be completed.

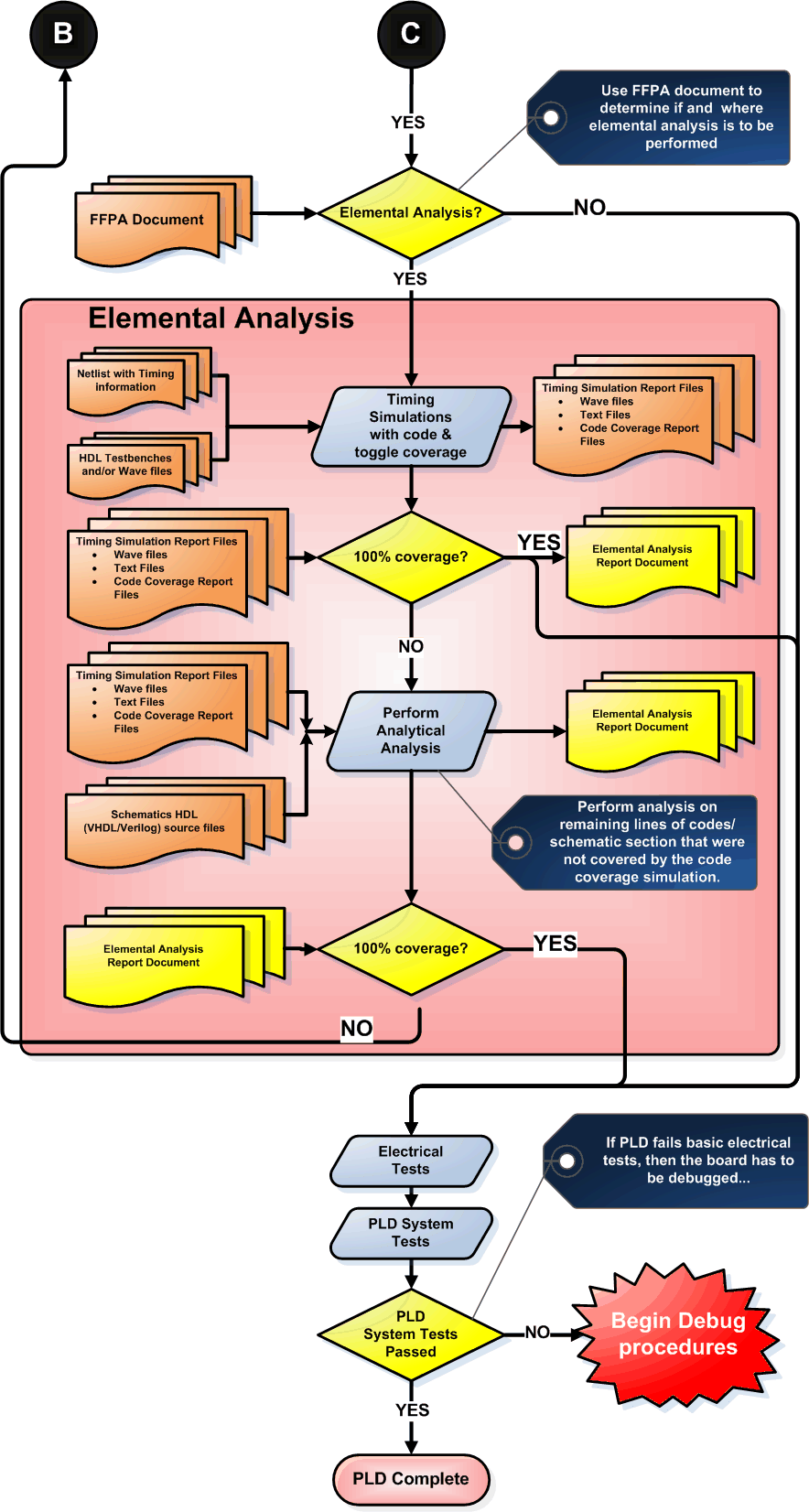
The verification can occur at any point in the design process, but for certification credit, a formal verification procedure shall be used on a configured item. For every test, there is information that shall be documented. It includes the following: the requirement or requirements being tested, test conditions and steps, pass/fail criteria, identification of test equipment used, and the configuration identity of the hardware item under test. Appendixes E-H are test forms that shall be used for verification cases and procedures. These forms have placeholders for the needed information. There are four different types of tests that may be conducted. They are functional test, HDL simulation, in-circuit test and analysis.

* Functional Test- This is the preferred method because it comes closet to simulating “real world” conditions, and is less labor intensive. These tests will involve having test software run in the processor to exercise the hardware and output the results to then be verified either by a tool or manually. See Appendix E for a template of how to document these tests.
* HDL Simulation- The simulation method typically involves running an Active-HDL test bench script to execute a test. A test bench is the preferred way to test the design using HDL simulation. Each test bench should be set up to run multiple tests (typically around 10 may be best), as this will make the organization easier by reducing the number of test bench setups needed. For the timing simulation, the maximum propagation delay shall be used, to ensure the test is covering the worst case scenario implementation. See Appendix F for a template of how to document these tests.
* In-Circuit Test- In circuit test involves using lab test equipment (such as oscilloscope) to accurately verify a measurement. See Appendix G for a template of how to document these tests.
* Analysis- Analysis may include a detailed examination of the functionality, performance, traceability and safety implications of a hardware item function and its relationship to other functions. Analysis activities alone or in combination with other verification methods provide evidence that a requirement is correctly implemented. Analysis should be based on data provided by the design process or other available databases. See Appendix H for a template of how to document the analysis.

The following Design Flow is repeated from the hardware development planning section of the PHAC and details the planned complex electronic hardware design-specific lifecycle. In this plan we will be focusing in on the Verification and Validation activities as it relates to this life cycle.







## 2.1 Review Methods

Verifying the satisfaction of the functional and performance requirements will be achieved by review, analysis of outputs, coverage analysis, and test. Review of test data is a formal verification by examining data outputs to determine that they are generated in the proper format and at the appropriate point in time. Logical path analysis is an example of an analysis technique. Tests will indicate that the correct path is taken through the code in response to the input values specified. Timing verification is also covered by review of test data. Tests will be conducted to show that the program executes within the allowable time span.

Review methods will include the use of document review checklist and project review checklists. These checklists form an instrumental part of the objective evidence required to show compliance to the objectives of DO-254. The checklists are contained in the appendix of this document.

Reviews shall be conducted at the following points (or documents) of the design life cycle:

* Requirements
* Concept Design
* Detailed Design
* Verification Cases and Procedures
* Verification Results
* Implementation
* Production Transition

Six characteristics for life cycle shall be evaluated in all reviews performed. They include:

**Unambiguous**

Information/data is written in terms that allow only a single interpretation.

**Complete**

Information / data include necessary and relevant requirements and descriptive material, labeled figures, and defined terms and unit of measure.

**Verifiable**

Information / data can be checked for correctness by a person or a tool.

**Consistent**

Information/data contains no conflicts.

**Modifiable**

Information/data is structured so that changes can be made completely, consistently, and correctly while retaining the structure.

**Traceable**

Information/data origin can be determined.

Every review will have an associated checklist of items to review against. The requirements review shall use the “Requirements Review Checklist” (Appendix A). Both the conceptual and the detailed design reviews shall use the “Design Review Checklist” (Appendix B). Both the verification cases and procedures and verification test results shall use the “Verification Cases, Procedures and Results Review Checklist” (Appendix C). The implementation review shall use the “Implementation Review Checklist” (Appendix I). The production transition review shall use the “Production Transition Review Checklist” (Appendix D).

If there are any defects, each one will be analyzed and a decision made whether it requires action to resolve it. The decision for each defect will be recorded in the review documentation and the resolutions will be independently verified. If a defect cannot be corrected within the

realm of the review, then a SCR must be entered into PVCS Tracker, and the SCR number must

be recorded in the review documentation.

Review of the Detail Design representations shall ensure that Date/Time stamp and revision label have been updated and are unique. Further, review of the detail design shall ensure that the Standards specified (*hardware Design Standards*) are met. HDL source code generated by Active-HDL from block diagrams shall be reviewed for correct output.

The verification results, including analysis, simulation, requirements coverage, and detail design coverage shall be reviewed. The verification results review shall ensure that the required coverage has been attained for both the hardware requirements data and detail design representations. Review of test results shall ensure that all actual results meet expected results and that any discrepancies are satisfactorily explained.

### 2.1.1 Review Verification and validation plans

Early in the review process, the Hardware Verification & Validation Plan is reviewed to ensure that activities planned for achieving test coverage, if followed, will satisfy the DO-254 objective. Other plans including the Plan for Hardware Aspects of Certification, Hardware Configuration Management Plan, Hardware Quality Assurance Plan, and tool plans (if applicable) may contain additional information related to test coverage.

The following questions are considered when reviewing the plans:

* Are the plans sufficiently clear and detailed to allow the development and quality engineers to follow them consistently?
* Do the plans specify who is allowed to perform verification tasks?
* Do the plans specify how each requirement will be tested (e.g., module test, Hardware integration, etc.)?
* Do the plans address all aspects of test coverage analysis? For example, the following are addressed:
* tools and tool qualification, if tools are used for test coverage
* the relationship between requirements-based testing and measuring test coverage
* a process for determining when additional requirements-based tests should be added, if coverage is not achieved as expected
* a procedure for regression analysis and testing, if necessary
* the transition criteria to start and end test coverage
* Do the plans address the Hardware change process for the airborne Hardware?
* Do the plans address regression analysis and testing with respect to the unique requirements for test coverage?
* Do the plans address possible reuse of verification tools? For example, is credit being claimed from previous tool qualifications or will the tool qualification data be used in a future program?
* Is there evidence that the plans are being followed (such as, progress against timeframes, staffing, verification records, and HPA records)?

### 2.1.2 Determine Need for Tool Qualification

When development and/or verification tools are used, they are documented in the Tool Qualification Plan and/or Plan for Hardware Aspects of Certification, as required. The following questions are considered when determining whether a tool needs to be qualified:

* Can the tool allow an existing error to remain undetected?
* Will there be little or no verification of the output of the tool?
* Will the tool be used to assist or replace a process that has a significant effect on the integrity of the product being developed?

### 2.1.3 Review Data Related To Qualification Tools

If tools are used for test coverage analysis, the following questions are considered when reviewing the qualification data:

* Do the plans state which test coverage tools are being qualified and the rationale for qualification?
* Are the specific tool requirements documented? DO-254, section 12.2.3.2 lists the typical information that should be included in the Tool Operational Requirements document.
* Is the effect of various coding considerations addressed?
* Does the tool qualification data address whether the tool needs to instrument the code to perform test coverage analysis?
* If the tool needs to instrument the code, has the effect of the instrumentation on the code been assessed?
* If the tool measures coverage at the object code level, is additional analysis available to support the equivalence of coverage at the object and source code levels?
* Is the tool qualification process sufficient to discover errors in the tool and limitations of the tool’s functions?
* Does the tool qualification data address how tool deficiencies should be handled if they are found while the tools are being used in a certification project?
* Does the tool qualification data detail how changes to the tool will be evaluated and controlled?
* Are procedures for using each tool documented?
* Are limitations of the tool clearly documented and addressed (e.g., the limitations discussed in chapter 4) that may affect assessment of coverage?
* Is the tool configuration controlled and documented in the plans and Hardware Life Cycle Environment Configuration Index?
* Are the quality engineers using the tool configuration identified in the plans and the Hardware Life Cycle Environment Configuration Index?

### 2.1.4 Review Test Cases and Procedures

During the verification process, quality (along with select members of the development team) reviews the requirements-based test cases to assure that all requirements are adequately covered. If the requirements-based tests are not adequate to achieve test coverage, then additional requirements-based tests or analyses may be needed.

The following questions are considered when evaluating test cases and procedures:

Do the test cases and procedures adhere to the relevant plans and standards?

* If plans or standards have not been followed, is there documented rationale for deviations from stated plans and standards?
* Is the rationale for each test case clearly explained?
* Are the test cases and procedures appropriately commented to allow future updates?
* Have the test cases and procedures been subjected to appropriate change and configuration control?
* Is the separation between test cases clear? For example, are test starts and stops identified?
* Do the test cases and procedures specify required input data and expected output data?
* Were the inputs for each test case derived from the requirements?
* Are the test cases and procedures sufficient to cover all the relevant requirements? That is, do the traceability matrices provide clear association between test cases and requirements?
* Are the test cases and procedures sufficient to achieve test coverage?
* Are sufficient tests to provide test coverage identified for each logic construct?
* Are there sufficient robustness test cases and procedures?
* Are test cases and procedures correct?

### 2.1.5 Review Checklists for Test Cases, Procedures, and Results

A checklist is used during review of test procedures and results (See Document Review Management System in the Hardware Quality Assurance Plan). During this review, the checklists themselves are assessed, considering the following questions for test coverage:

* Are the checklists sufficient to determine that the requirements-based test cases, procedures, and results meet the test coverage objective?
* Have the checklists been prepared and/or reviewed by quality?
* Do the checklists specify:
* who performed the review?
* what data was reviewed (with revision)?
* when it was reviewed?
* what was found?
* what corrective actions were taken, if necessary?
* Do the checklists require evaluation of tolerances specified in the requirements?
* Do the checklists ensure that results of the test cases can be visually verified? (e.g., Can the HPA, or reviewer, visually determine when requirements-based tests have passed or failed?)
* Will the checklists reveal whether the results of the test cases that are counted for credit towards test coverage are observable?
* Will the checklists address limitations of the structural coverage analysis tool as documented in the tool qualification?
* Will the checklists reveal test cases that violate project standards?

### 2.1.6 Effectiveness of the Test Program

HPA relies on the following tasks to determine the effectiveness of the overall test program.

#### 2.1.6.1 Assess Results of Requirements-Based Tests

The first step after test execution is to determine whether all requirements-based tests pass. In addition to checking the final pass/fail results, the test cases and results for some randomly selected requirements should be examined to ensure that the results reflect the given inputs for those cases. Test results are also checked carefully with respect to any specified tolerances.

The following questions are considered to assess the requirements-based test results:

* Are the test result files clearly linked to the test procedures and codes?
* Are failed test cases obvious from the test results?
* Do the test results indicate whether each procedure passed or failed and the final pass/fail results?
* Do the test results adhere to the relevant plans, standards, and procedures?
* Have the test results been subjected to appropriate configuration control?

#### 2.1.6.2 Assess Failure Explanations and Rework

Each failed test case is documented with an explanation for why it failed, including references to applicable Action Request. In some cases, rework of some life cycle data will be required; in other cases, only an explanation for the failed test cases is needed. If rework is required, the impact of changes should be carefully evaluated and the changed items should be subjected to the appropriate change and configuration control.

Once all rework is complete, test cases should be rerun in compliance with plans for regression testing. Note: there may be cases where failed requirements-based tests are acceptable; however, it is typical for them to be fixed and rerun.

The following questions are considered to assess failures and rework:

* Is there an acceptable rationale for deviations from expected results, standards, or plans?
* Are explanations for the failed test cases technically sound and accurate?
* Do explanations for failed test cases contain accurate references to relevant problem reports?
* Are explanations for code or test rework suitable to address the failure?
* Have test cases been re-executed in compliance with plans for regression testing?
* Have the test results from regression testing been documented appropriately?

#### 2.1.6.3 Assess Elemental Analysis coverage achievement

In general, HPA produces test cases that are expected to achieve 100% test coverage (i.e., the purpose of test documentation is to show compliance with all of the requirements). If all the requirements have been covered by tests without achieving full test coverage, dead code, unintended functionality, or incorrectly documented, de-activated code may be indicated. It is the policy to remove all dead code.

The following questions are considered when assessing coverage achievement:

* Has the test coverage criteria been correctly applied?
* Is 100% test coverage achieved through requirements-based testing?
* If 100% test coverage is not achieved through requirements-based testing, is there an explanation detailing which parts of the code were not executed, and why? Have additional test cases been added?
* Are explanations for drops in coverage sufficiently detailed and acceptable?
* Are there problem reports associated with dead code?
* Has dead code been analyzed and/or removed?

## 2.2 Analysis Of Outputs Methods

The analysis of outputs methods are specific to each analysis being performed. The subsequent paragraphs detail the methods which will be used for each analysis performed as part of the Hardware verification process.

### 2.2.1 System And Hardware Requirements Trace Analysis Method

The results of this analysis will be contained in the system requirements trace matrix. This requirements trace matrix will be constructed in accordance with the rules of the Hardware Requirement Standard, 800-HRS-01, and as follows

* The requirement identifier for each system requirement allocated to Hardware will be entered into one field of the matrix.
* The requirement identifier for each Hardware requirement that satisfies the system requirement will be entered into the other field of the matrix.
* When multiple Hardware requirements satisfy one system requirement, an entry with the duplicate system requirement identifier field will be entered.
* When multiple system requirements are satisfied by one Hardware requirement, an entry with the duplicate Hardware requirement identifier field will be entered.
* All Hardware requirements derived due to implementation will be designated as “Derived” in the system requirement identifier field.

The requirements trace matrix will be used to identify any system requirements allocated to Hardware which are not being addressed in the Hardware requirements.

### 2.2.2 Hardware Requirements Specification Trace Analysis Method

The results of this analysis will be contained in the Hardware requirements trace matrix. This requirements trace matrix will be appended as follows:

* The name of each CSU designed in the Hardware will be entered into a field of the matrix.
* Each Hardware requirement identifier previously entered in the matrix will be assigned to the appropriate CSU that by design satisfies the requirement.
* When multiple CSUs satisfy one Hardware requirement, an entry with the duplicate Hardware requirement identifier field will be entered.
* When multiple Hardware requirements are satisfied by one CSU an entry with the duplicate CSU name field will be entered.

The requirements trace matrix will be used to identify any Hardware requirements that are not being addressed in the Hardware architecture.

### 2.2.3 Requirements-Based Test Coverage Analysis Method

The results of this analysis will be contained in the requirements verification matrix.

* An identifier for each Hardware test case will be entered into a field of the matrix.
* Each Hardware requirement identifier previously entered in the matrix will be assigned to the appropriate test case that verifies the requirement.
* When multiple test cases satisfy one Hardware requirement, an entry with the duplicate Hardware requirement identifier field will be entered.
* When multiple Hardware requirements are verified by one test case an entry with the duplicate test case identifier field will be entered.
* Evaluate the capability of each test case to adequately verify the corresponding Hardware requirement.

The requirements trace matrix will be used to identify any Hardware requirements that are not being addressed in the Hardware architecture.

### 2.2.4 Hardware Reviews and Analysis

Reviews and analysis are applied to the results of the Hardware development process and Hardware verification process. A distinction is made between the term reviews and the term analysis. Analysis provides repeatable evidence of correctness and reviews provide a qualitative assessment of correctness. A review may consist of an inspection of an output of a process guided by a checklist or similar aid. An analysis may examine in detail the functionality, performance, traceability, and safety implications of a Hardware component, and its relationship to other components within airborne system or equipment.

#### 2.2.4.1 Reviews and Analysis of High-Level Requirements

The objective of these reviews and analysis is to detect and report requirements errors that may have been introduced during the Hardware requirements process. These reviews and analysis confirm that the high-level requirements satisfy these objectives:

* Compliance with system requirements: The objective is to ensure that the system functions to be performed by the Hardware are defined, that the functional, performance, and safety-related requirements of the system are satisfied by the Hardware high-level requirements, and that derived requirements of the system are satisfied by the Hardware high-level requirements, and that derived requirements and the reason for their existence are correctly defined.
* Accuracy and consistency: The objective is to ensure that each high-level requirement is accurate, unambiguous, and sufficiently detailed and that the requirements do not conflict with each other.
* Compatibility with the target computer: The objective is to ensure that no conflicts exist between the high-level requirements and the system features of the target computer, especially, system response times and input/output hardware.
* Verifiability: The objective is to ensure that each high-level requirement can be verified.
* Conformance to standards: The objective is to ensure the Hardware Requirements Standards were followed during the Hardware requirements process and that deviations from the standards are justified.
* Traceability: The objective is to ensure that the functional, performance, and safety-related requirements of the system that are allocated to Hardware were developed into the Hardware high-level requirements.
* Algorithm aspects: The objective is to ensure the accuracy and behavior of the proposed algorithms, especially in the area of discontinuities.

#### 2.2.4.2 Reviews and Analysis of Low-Level Requirements

The objectives of these reviews and analysis is to detect and report requirements errors that may have been introduced during the Hardware design process. These reviews and analysis confirm that the Hardware low-level requirements satisfy these objectives:

* Compliance with high-level requirements: The objective is to ensure that the Hardware low-level requirements satisfy the Hardware high-level requirements and that derived requirements and the design basis for their existence are correctly defined.
* Accuracy and consistency: The objective is to ensure that each low-level requirement is accurate and unambiguous and that the low-level requirements do not conflict with each other.
* Compatibility with the target computer: The objective is to ensure that no conflicts exist between the Hardware requirements and the system features of the target computer, especially, the use of resources (such as bus loading), system response times, and input/output hardware.
* Verifiability: The objective is to ensure that each low-level requirement can be verified.
* Conformance to standards: The objective is to ensure that the Hardware Design Standards were followed during the Hardware design process, and that deviations from the standards are justified.
* Traceability: The objective is to ensure that the high-level requirements and derived requirements were developed into the low-level requirements.
* Algorithm aspects: The objective is to ensure the accuracy and behavior of the proposed algorithms, especially in the area of discontinuities.

#### 2.2.4.3 Reviews and Analysis of Hardware Architecture

The objective of these reviews and analysis is to detect and report errors that may have been introduced during the development of the Hardware architecture. These reviews and analysis confirm that the Hardware architecture satisfies these objectives:

* The compatibility with the high-level requirements
* Consistency
* Compatibility
* Verifiability
* Conformance to standards
* Partitioning integrity

#### 2.2.4.4 Reviews and Analysis of Outputs of the Integration Process

The objective is to ensure that the results of the integration process are simple and correct.

#### 2.2.4.5 Reviews and Analysis of Test Cases, Test Procedures and Results

The objective of these reviews and analysis is to ensure that the testing of the hardware was developed and performed accurately and completely. The topics should include:

* Test cases: Verification of test cases is presented later in this document.
* Test procedures: The objective is to verify that the test cases were accurately developed into test procedures and expected results.
* Test results: The objective is to ensure that the test results are correct and that discrepancies between actual and expected results are explained.

## 2.3 Coverage Methods

The subsequent paragraphs detail the methods that will be used for coverage analysis as part of the Hardware verification process.

Coverage refers to the extent to which a given verification activity has satisfied its objectives. Coverage analysis measures will be applied to both requirements definition and testing activities. Appropriate coverage measures will be used to manage and audit verification activities. This will aid in determining the adequacy of the verification accomplished.

Coverage is viewed as a measure, not a method or a test. As such, results will be expressed as the percentage of an activity that is accomplished. Two specific measures of test coverage are identified in the following figure: requirements coverage and Hardware structure coverage.

Requirements coverage analysis will be used to determine how well the requirements-based testing verifies the implementation of the Hardware requirements and establishes traceability between the Hardware requirements and the test cases. Structural coverage analysis will be used to determine how much of the code structure was executed by the requirements-based tests and establishes traceability between the code structure and the test cases.

### 2.3.1 Requirements Coverage

Each hardware requirement contains a finite list of behaviors and features, and that each requirement is written to be verifiable. Testing based on requirements will be performed from the perspective of the user (providing a demonstration of intended function), and will provide a means for the development of test cases concurrently with development of the requirements.

In addition, Hardware requirements may be created that cannot be determined from top-level Hardware specifications. Derived requirements, as described in DO-254, will be used for this reason. Derived requirements will be tested as part of requirements-based testing.

### 2.3.2 Simulation

Simulation for credit shall be performed on a HDL timing model described in *Design Standards*. Simulation shall be performed using a HDL Simulator. Many requirements may be verified with each test case. As much as possible, HDL simulation of the design shall be done at the “chip” level, or “at the pins.”

This testing shall be done in a HDL testbench environment which instantiates the design. The simulation should not provide stimulus access to signals within the design. The testbench should contain behavioral models that accurately simulate the behavior of devices that interface to the PHD. The behavioral models within the testbench shall use back-channel communication to pass expected data values, etc. between themselves to ensure proper operation in the simulation. Primary direction of the simulation will be provided through a processor behavioral model.

The HDL behavioral models in the testbench shall indicate any failure conditions and halt simulation. Otherwise, the HDL simulation procedure shall indicate successful completion. The verification result for a test case consists of the pass/fail indication provided by the test case and the saved simulation waveform from executing the test.

For each new HDL timing model produced in the Implementation step, all simulation procedures shall be run again. The implementation can greatly vary the position of blocks and routing of signals for seemingly minor changes, which can lead to great variations in the timing parameters in the model.

### 2.3.3 Static Timing Analysis

Most hardware Place and Route tools have some level of static timing analysis included with them. When hardware place & route is done, the tool generates a report that can be used for static timing analysis. The designer shall check the frequency of the main clock against the maximum operative frequency from the report and ensure that the design has a minimum 10% margin. Synchronous logic design coupled with static timing analysis provides a reliable method to verify robust timings in overall design. The static analysis report should be included as part of the verification results document.

### 2.3.4 Post-Route Simulation

Inputs

* HDL Timing Simulation Model
* Actual signal delay data based on synthesized design and device models
* Simulation Tool
* Simulation input vectors, test benches, and/or other stimulus file types created in detail design step

Outputs

* Design corrections needed based on the results of the simulation

The delay data is back-annotated into the design database and then the hardware design simulation is rerun to verify that the design still meets its performance requirements and all required timing margins. In post-route simulation, the maximum delay should be used for timing analysis, while simulation with minimum or typical delay is possible. The maximum-delay simulations shall include both on-chip (clock speed, propagation delay) timing verification and off-chip (set-up, clock-to-out) timing verification. If the simulation passes, the process proceeds to the Program and Test step. If the simulation fails, corrections shall be made and the appropriate process steps shall be repeated. The steps that shall be repeated may start at the Initial Design step, Synthesis step, or Place and Route step, depending on the nature and severity of the design corrections. Varying the programmable logic performance parameters until the programmable logic simulation fails may also perform margin analysis. This would provide data on how much margin exists above the frequency at which the programmable logic device was designed to operate. The simulations performed in this step may be repeated in the formal hardware verification process.

2.4 Testing Methods

Verification Testing of Hardware has two objectives. One objective is to demonstrate that the Hardware satisfies its requirements. The second objective is to demonstrate with a high degree of confidence that errors that could lead to unacceptable failure conditions, as determined by the system safety assessment process, have been removed. Three types of testing are utilized:

* System Regression Testing: To verify correct operation of the Hardware in the target environment.
* Hardware Verification Testing: To verify the interrelationships between Hardware requirements and components and to verify the implementation of the Hardware requirements and Hardware components within the Hardware architecture.
* Production Testing: To verify the implementation of Hardware for every unit.

To satisfy the Hardware testing objectives:

* Test cases are developed based primarily on the Hardware requirements.
* Test cases are developed to verify correct functionality and to establish conditions that reveal potential errors.
* Hardware requirements coverage analysis is used to determine what Hardware requirements were not tested.

### 2.4.1 Test Environment

More than one test environment may be needed to satisfy the objectives for Hardware testing.

#### 2.4.1.1 Requirements-Based Test Cases

Requirements-based testing is emphasized because this strategy has been found to be the most effective at revealing errors. Requirements-based test case selection includes the following:

* Implementation of both normal range and robustness (abnormal range) test cases.
* The specific test cases should be developed from the Hardware requirements and the error sources inherent in the Hardware development process.

#### 2.4.1.2 Normal Range Test Cases

Normal Range test cases are developed to demonstrate the ability of the Hardware to respond to normal inputs and conditions.

#### 2.4.1.3 Robustness Test Cases

Robustness test cases are developed to demonstrate the ability of the Hardware to respond to abnormal inputs and conditions.

#### 2.4.1.4 Requirements-Based System Verification Testing Methods

Requirements-based system verification testing methods concentrate on error sources associated with the Hardware operating within the target environment, and on the high-level functionality. The objective of requirements-based testing is to ensure that the Hardware will satisfy the high-level requirements.

# Verification Data

This section includes identification and description of the evidence to be produced as a result of the hardware verification process.

Reference: See DO-254 Section 10.1.4 Objective 2.

Review forms become quality assurance records, under HC2 control, managed in accordance with the Configuration Management Process. Each test case may have several files of different types associated with it. All test cases and procedures, along with any analysis required, will be released as *Verification Cases and Procedures*. The success or failure indications, simulation waveforms, detail design coverage data, etc. shall be saved as *Verification Results*. The Problem Reports generated from verification activities will be handled in accordance with the Configuration Management Process.

# Verification Independence

Independence is achieved through the “No Sole Perspective” method. This perspective proposes that that there is value in having someone other than the developer of the data review the data, and that it satisfies the criteria for having an “objective evaluation” without requiring organizational independence. In fact, this perspective recommends that there is additional benefit in having multiple other persons involved in each review from different disciplines (such as systems engineers, safety specialists, test engineers, human factors specialists, technical writers, etc.). Also, by having other disciplines involved in the review, one could potentially be getting the greatest possible “objective evaluation” of the data. Independent reviews help prevent a biased perspective since it may be difficult to impartially review one’s own work.

Additionally, the value of having an independent reviewer involved in the software engineering discipline is supported by extensive research and application. It is also intuitive and reasonable that having someone other than the author or developer of an artifact, review (inspect) that artifact from their different perspectives, disciplines, and experiences will provide for higher quality, safer, easier to maintain, and less expensive (in the long run) products.

This project expands on the “No Sole Perspective,” and proposes the following guidelines:

1. General Position: To achieve verification independence, the person performing or responsible for the verification activity will not be the same person who developed the data being verified.
2. Tool Qualification: If a tool is used to eliminate, reduce or automate the activities associated with a DO-254 objective needing verification independence and that tool’s output will not be completely verified with independence, then that tool will be qualified.
3. Test Case and Procedure Development: The test cases and procedures will not be developed by the same person who developed the low-level requirements or source code to be verified by those test cases and procedures.
4. Test Case and Procedure Review: The person responsible for performing the test cases and procedures review will not be the same person who developed the test cases and procedures to be verified.
5. Test Execution: The person responsible for executing the tests will not be the same person who developed the requirements or code being verified by the tests, nor the developer of the test cases and procedures being executed. If the test execution is fully automated (e.g., scripted “batch” run with no need for human intervention or observation), then this guideline would not apply. However, that test “tool” may need to be qualified and the developer of the testing tool (that person setting up the automated test execution and environment) will not be the same person who developed the test cases and procedures.
6. Test Results Review and Coverage Analysis: The person responsible for performing the test results review or test coverage analyses will not be the same person who developed the test cases and procedures, or the same person who executed the tests.

# Verification Environment

This section includes identification and description of analysis and test equipment and verification tools to be used to implement the verification process and activities.

The verification environment consists of various equipment, forms, and procedures, which vary depending on the type of verification that is used. The actual simulation/test environment (including appropriate tool identification and calibration information) used shall be recorded in the verification results.

## 5.1 Tools

Refer to the Plan for Hardware Aspects of Certification for the tool qualification justification.

**Development and Verification Tools**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Tool** | **Version** |
| Design Entry | Aldec Active HDL,  Mentor Graphics ViewLogic | 6.3  V7.0 |
| Synthesis | Xilinx ISE | 6.3g |
| Place & Route | Xilinx ISE | 6.3g |
| Simulation | Active HDL | 4.0 |
| Device Programming | Xilinx iMPACT | 6.3.0.1i |
| Source Revision Control | Visual Source Safe | 6.0 |

## 5.2 Functional Tests

Functional tests will be conducted on the actual hardware that is being verified. Test scripts running on the processor will stimulate the processor to achieve the verification objective.

## 5.3 HDL Simulation

HDL simulation tests will run using Aldec Active HDL on an IBM PC compatible Computer with windows XP operating system.

## 5.4 In-Circuit Test

In-circuit tests will use lab equipment such as multimeter, oscilloscope, and logic analyzer to achieve the verification objective.

## 5.5 Analysis Test

Analysis will use an IBM PC compatible computer with a Windows XP operating system to view the necessary data to achieve the verification objective.

# Appendix A: Requirements Review Checklist

Requirements Review Checklist

**Accurate and Consistent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| R1 | Derived requirements are consistent with the system requirements or requirements from which it is derived |  |  |  |
| R2 | Requirements are compatible with relevant hardware design standards. |  |  |  |
| R3 | Components requirements (performance, temp range, de-rating, screening) are consistent with the safety and reliability requirements |  |  |  |
| R4 | Requirements are traceable upward to the next hierarchical level. |  |  |  |

**Verifiable**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| R5 | Requirements are unambiguous, verifiable, and described in complete enough detail for its hierarchical level and does not conflict with other requirements. |  |  |  |
| R6 | Requirements are compatible with the capabilities and limitations of available technology |  |  |  |

**Understandable/Complete**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| R7 | Software/Hardware interface requirements are defined |  |  |  |
| R8 | Derived requirements capture the implementation constraints that will not be verified at a higher hierarchical level. |  |  |  |

**General**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| R9 | Ability to test, maintain and manufacture the hardware item is addressed |  |  |  |
| R10 | Requirements are consistent with the System Safety Assessment. |  |  |  |
| R11 | Derived safety requirements are fed back to the System Safety Assessment for validation. |  |  |  |

# Appendix B: Design Review Checklist

Design Review Checklist

**General**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| D1 | All requirement are addressed and the derived requirements and the design data are correctly defined |  |  |  |
| D2 | Environmental requirements are addressed |  |  |  |
| D3 | Safety and reliability requirements are addressed |  |  |  |
| D4 | Safety aspects of the design data are explicitly identified and validated. |  |  |  |
| D5 | The design is capable of being implemented, tested and maintained |  |  |  |
| D6 | New manufacturing techniques have been evaluated |  |  |  |
| D7 | Component selection criteria identified in the plans has been satisfied. |  |  |  |
| D8 | Design is traceable to the requirements |  |  |  |
| D9 | Does the Design conform to the design standards |  |  |  |
| D10 | Does the Code conform to the Coding standards |  |  |  |

# Appendix C: Verification Cases, Procedures, and Results Review Checklist

Verification Cases, Procedures, and Results Review Checklist

**General**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| T1 | Each requirement to be validated or verified by test is identified. (for results review, ensure each identified requirement HAS been verified) |  |  |  |
| T2 | Testing stimulus, sequence and test conditions, such as ambient temperature and applied voltage, are defined for each test |  |  |  |
| T3 | Pass/Fail criteria and a method for recording the results are defined prior to test execution. |  |  |  |
| T4 | The verification of each requirement is appropriate and complete, especially with respect to safety requirements |  |  |  |
| T5 | Complete identification of the test equipment and calibration date for each is recorded |  |  |  |
| T6 | The configuration identity of the hardware item being tested is recorded |  |  |  |
| T7 | Test results are recorded and retained (failure/discrepancy documented in SCR) |  |  |  |

# Appendix D: Production Transition Review Checklist

Production Transition Review Checklist

**General**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** |  |  |  |
| P1 | Manufacturing data is prepared from configured design data |  |  |  |
| P2 | Manufacturing data is checked for completeness and consistency with the configured design data |  |  |  |
| P3 | Any changes or improvements that are incorporated during the production transition process are evaluated to ensure they adhere to all product requirements, especially safety. |  |  |  |
| P4 | Manufacturing requirements pertaining to safety are explicitly defined so they can be controlled during the production process |  |  |  |
| P5 | Data required to develop acceptance test criteria is determined. |  |  |  |

# Appendix E: Functional Test Standard

**[LRU under test] [unique test #]**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Objective:** | [*objective of test*] | | | | | | |
| **Requirement(s) Tested:** | HLR #s [*115* | | | | | | |
| **Tester(s):** |  | **Test Date:** |  | | | | |
| **SW Version :** |  | **CCA Revision:** |  | | | | |
| **HW Version:** |  |  |  | | | | |
|  | | | | | | | |
| **Test Procedure:** | | | | | | | |
| 1. Run test file PHD\_XXXX.c…   2) ….  3)…  4)…..  5).. | | | | | | | |
| **Pass/Fail Criteria:** | | | | | | | |
| Results match with results in PHD\_results.txt file | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| **Explain Discrepancies:** | | | | | | | |
|  | | | | | | | |

# 

# Appendix F: HDL Simulation Test Standard

**[LRU under test] [unique test #]**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Objective:** | [*objective of test*] | | | | | | |
| **Requirement(s) Tested:** | HLR #s [*115*] | | | | | | |
| **Tester(s):** |  | **Test Date:** |  | | | | |
| **HW Version:** |  |  |  | | | | |
| **Simulation Tools:** | Active HDL 6.2 | [*other tools*] | [*other tools*] | | | | |
|  | | | | | | | |
| **Test Procedure:** | | | | | | | |
| 1. Run test file PHD\_XXXX.do… 2. …. 3. … 4. ….. 5. .. | | | | | | | |
| **Pass/Fail Criteria:** | | | | | | | |
| Output signal XXX is high for greater than 1ms. | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| **Explain Discrepancies:** | | | | | | | |
|  | | | | | | | |

# Appendix G: In-Circuit Test Standard

**[LRU under test] [unique test #]**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Objective:** | [*objective of test*] | | | | | | |
| **Requirement(s) Tested:** | HLR #s [11*5*] | | | | | | |
| **Tester(s):** |  | **Test Date:** |  | | | | |
| **SW Version :** |  | **CCA Revision:** |  | | | | |
| **PHD Version:** |  |  |  | | | | |
| **Test Equipment ID Number and Calibration Date** | Oscilloscope  Serial # 123456  12/31/2006 | [*other equipment*] | [*other equipment*] | | | | |
|  | | | | | | | |
| **Test Procedure:** | | | | | | | |
| 1. Connect a scope to the output of pin #...... 2. …. 3. … 4. ….. 5. .. | | | | | | | |
| **Pass/Fail Criteria:** | | | | | | | |
| Pin #xx of the HW is at a logic high  And pin #yy of the HW is a logic low | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| **Explain Discrepancies:** | | | | | | | |
|  | | | | | | | |

# 

# Appendix H: Analysis Standard

**[LRU under test] [unique test #]**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Analysis Objective:** | [*objective of analysis*] | | | | | | |
| **Requirement(s) Analyzed:** | HLR #s [11*5*] | | | | | | |
| **Tester(s):** |  | **Test Date:** |  | | | | |
| **Version of Analyzed Item :** | [*schematic #XXXXX rev B]* | **Version of Analyzed Item:** | [*12345BOM revC]* | | | | |
| **Version of Analyzed Item:** |  | **Version of Analyzed Item** |  | | | | |
|  | | | | | | | |
| **Analysis Procedure:** | | | | | | | |
| 1. Inspect schematic #XXXXX verify connection of pin 20 on processor…..   2) ….  3)…  4)…..  5).. | | | | | | | |
| **Pass/Fail Criteria:** | | | | | | | |
| Pin 20 is connected to ……. | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| [*other test criteria*] | | | | | | | |
|  | | | | Pass |  | Fail |  |
| **Explain Discrepancies:** | | | | | | | |
|  | | | | | | | |

# Appendix I: Implementation Review Checklist

Implementation Review Checklist

**General**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Criteria** | **Y** | **N** | **NA** |
| I1 | A hardware item has been produced which implements the hardware detailed design using representative manufacturing processes |  |  |  |
| I2 | The hardware item implementation, assembly and installation data is complete |  |  |  |
| I3 | Derived requirements are fed back to the detailed design process or other appropriate processes. |  |  |  |
| I4 | Requirement omissions and errors are provided to the appropriate process for resolution. |  |  |  |
| I5 | Test cases and procedures are under CM |  |  |  |
| I6 | Test results are under CM |  |  |  |
| I7 | Action items are closed |  |  |  |
| I8 | Process assurance transition records exist |  |  |  |