DESIGN ASSURANCE GUIDANCE
FOR AIRBORNE ELECTRONIC HARDWARE

RTCA/DO-254	**Prepared by:**
April 19, 2000	**SC-180**

RTCA, Incorporated
1140 Connecticut Avenue, N.W., Suite 1020
Washington, DC 20036-4001 USA

DESIGN ASSURANCE GUIDANCE FOR AIRBORNE ELECTRONIC HARDWARE

RTCA/DO-254 **Prepared by: SC-180**

April 19, 2000

Copies of this document may be obtained from

RTCA, Inc.
1140 Connecticut Avenue, NW, Suite 1020
Washington, DC 20036-4001 USA

Telephone: 202-833-9339
Facsimile: 202-833-9434
Internet: www.rtca.org

Please call RTCA for price and ordering information.

FOREWORD

This document was prepared by RTCA Special Committee 180 (SC-180). It was approved by the RTCA Program Management Committee on April 19, 2000.

RTCA SC-180 and the European Organization for Civil Aviation Equipment (EUROCAE) WG-46 jointly accomplished the development of this guidance through the consensus process.

RTCA, Incorporated is a not-for-profit organization formed to advance the art and science of aviation and aviation electronic systems for the benefit of the public. The organization functions as a Federal Advisory Committee and develops consensus-based recommendations on contemporary aviation issues. RTCA's objectives include but are not limited to:

- Coalescing aviation system user and provider technical requirements in a manner that helps government and industry meet their mutual objectives and responsibilities.
- Analyzing and recommending solutions to the system technical issues that aviation faces as it continues to pursue increased safety, system capacity and efficiency.
- Developing consensus on the application of pertinent technology to fulfill user and provider requirements, including development of minimum operational performance standards for electronic systems and equipment that support aviation.
- Assisting in developing the relevant technical material upon which positions for the international Civil Aviation Organization and the International Telecommunication Union and other interested international organizations can be based.

The organization's recommendations are often used as the basis for government and private sector decisions as well as the foundation for many Federal Aviation Administration Technical Standard Orders.

Since RTCA is not an official agency of the United States Government, its recommendations may not be regarded as statements of official government policy unless so enunciated by the U.S. government organization or agency having statutory jurisdiction over any matters to which the recommendations relate.

EXECUTIVE SUMMARY

The development and use of complex electronic hardware by the aviation industry has created new safety and certification concerns. In response, RTCA SC-180 and EUROCAE WG-46 were formed. WG-46 and SC-180 agreed to become a joint committee early in the development of this document. This joint committee was chartered to develop clear and consistent design assurance guidance for electronic airborne hardware such that it safely performs its intended functions.

Electronic airborne hardware includes line replaceable units, circuit board assemblies, application specific integrated circuits, programmable logic devices, etc. This guidance is applicable to current, new, and emerging technologies.

The guidance in this document is intended to be used by aircraft manufacturers and suppliers of electronic hardware items for use in aircraft systems. The hardware design life cycle processes are identified. Objectives and activities for each process are described. The guidance is applicable to all hardware design assurance levels as determined by the system safety assessment.

In the development of this document, the committee considered other industry documents including Society of Automotive Engineers (SAE) Aerospace Recommended Practice (ARP) document ARP4754/EUROCAE ED-79, Certification Considerations for Highly Integrated or Complex Aircraft Systems; SAE ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment; and RTCA DO-178/EUROCAE ED-12, Software Considerations in Airborne Systems and Equipment Certification.

TABLE OF CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

The use of increasingly complex electronic hardware for more of the safety critical aircraft functions generates new safety and certification challenges. These challenges arise from a concern that said aircraft functions may be increasingly vulnerable to the adverse effects of hardware design errors that may be increasingly difficult to manage due to the increasing complexity of the hardware. To counteract this perceived escalation of risk it has become necessary to ensure that the potential for hardware design errors is addressed in a more consistent and verifiable manner during both the design and certification processes.

As airborne electronic hardware becomes more complex, technology evolves and experience is gained in the application and use of the procedures described in this document, this document will be revised and reviewed consistent with approved RTCA/EUROCAE procedures.

1.1 Purpose

This document has been prepared to assist organizations by providing design assurance guidance for the development of airborne electronic hardware such that it safely performs its intended function, in its specified environments. This guidance should be equally applicable to current, new, and evolving technologies. The purposes of this document are to:

1. Define hardware design assurance objectives.

2. Describe the basis for these objectives to help ensure correct interpretation of the guidance.

3. Provide descriptions of the objectives to allow the development of means of compliance with this and other guidance.

4. Provide guidance for design assurance activities to meet the design assurance objectives.

5. Allow flexibility in choice of processes necessary to meet the objectives of this document including improvements, as new process technologies become available.

This document recommends the activities that should be performed in order to meet design assurance objectives, rather than detailing how a design should be implemented.

The philosophy used to generate this guidance document is one of a top-down perspective based on the system functions being performed by electronic hardware and not a bottom-up perspective or one based solely on the specific hardware components used to implement the function. A top-down approach is more effective at addressing safety design errors by facilitating informed system and hardware design decisions, and efficient and effective verification processes. For example, verification should be performed at the highest hierarchical level of the system, assembly, and subassembly, component or hardware item at which compliance of the hardware item to its requirements can be achieved and the verification objectives satisfied.

1.2 Scope

This document provides guidance for design assurance of airborne electronic hardware from conception through initial certification and subsequent post certification product improvements to ensure continued airworthiness. It was developed based on showing compliance with certification requirements for transport category aircraft and equipment but parts of this document may be applicable to other equipment.

The relationship between the system life cycle and the hardware design life cycle is described to aid in the understanding of the interrelationships of the system and hardware design assurance processes. A complete description of the system life cycle, including system safety assessment (SSA) and validation, and the aircraft certification process is not intended.

Certification issues are discussed only in relation to the hardware design life cycle. Aspects concerning the ability to produce, test, and maintain the hardware item are addressed only as they relate to airworthiness of the hardware design.

The guidance in this document is applicable, but not limited to, the following hardware items:

1. Line Replaceable Units (LRUs).

2. Circuit Board Assemblies.

3. Custom micro-coded components, such as Application Specific Integrated Circuits (ASICs) and Programmable Logic Devices (PLDs), including any associated macro functions.

4. Integrated technology components, such as hybrids and multi-chip modules.

5. Commercial-Off-The-Shelf (COTS) components.

Additional considerations that refer specifically to COTS components are included in Section 11 since COTS component suppliers may not necessarily follow the design processes described by this document or provide the necessary hardware design life cycle data.

This document does not attempt to define firmware. Firmware should be classified as hardware or software and addressed by the applicable processes. This document assumes that during the system definition, functions have been allocated to either hardware or software. RTCA DO-178/EUROCAE ED-12 provides guidance for functions that are allocated to implementation in software. This document provides guidance for functions that are allocated to hardware.

Note: *This allows an efficient method of implementation and design assurance to be determined at the time the system is specified and functions allocated. All parties should agree with this system decision at the time that the allocation is made.*

Assessment and qualification of tools used for hardware item design and verification is addressed in Section 11.4.

This document does not provide guidance concerning organizational structures or how responsibilities are divided within those structures.

Environmental qualification criteria are also beyond the scope of this document.

1.3 Relationship to Other Documents

In addition to the airworthiness requirements, various national and international standards for hardware are available. In some communities, compliance with these standards may be required. However, it is outside the scope of this document to invoke specific national or international standards, or to propose a means by which these standards might be used as an alternative or supplement to this document.

Where this document uses the term “standards”, it should be interpreted to mean the use of project-specific standards as applied by the airborne system, airborne equipment, engine, or aircraft manufacturer. Such standards may be derived from general standards produced or adopted by the manufacturer. Guidance for standards is provided in Section 10.2.

1.4 Related Documents

SAE ARP4754/EUROCAE ED-79, Certification Considerations for Highly Integrated or Complex Aircraft Systems, as a source of development guidance for highly integrated or complex aircraft systems.

SAE ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, as a source of safety assessment methods to be used in the hardware design assurance process.

RTCA DO-178/EUROCAE ED-12, Software Considerations in Airborne Systems and Equipment Certification, as the complementary document for software development assurance.

RTCA DO-160/EUROCAE ED-14, Environmental Conditions and Test Procedures for Airborne Equipment, may be used by equipment designers as the primary environmental test standard for hardware item qualification.

1.5 How to Use This Document

This document is intended to be used by the international aviation community. To aid such use, references to specific national regulations and procedures are minimized. Instead, generic terms are used. For example, the term “certification authority” is used to mean the organization or person granting approval on behalf of the country responsible for certification. Where a second country or a group of countries validates or participates in this certification, this document may be used with due recognition given to bilateral agreements or memoranda of understanding between the countries involved.

The guidance in this document represents a consensus of the aviation community and is a collection of the best industry practices for design assurance of airborne electronic hardware. To take into account the process developed in this document, the intent was to produce guidance that should be applied to complete new hardware designs and subsequent changes. Guidance for hardware previously developed to other processes is addressed in Section 11.1. It is understood that means other than those described herein may be available to and be used by the applicant.

In cases where examples are used to indicate how the guidance might be applied, either graphically or through narrative, the examples are not to be interpreted as the preferred method.

Section 11 discusses additional considerations for specific known cases where some of the objectives of Section 2 through Section 9 may not be satisfied. These considerations include guidance for the use of previously developed hardware, COTS component usage, product service experience, and tool assessment and qualification.

Appendix A provides guidance for the necessary hardware design life cycle data based on the hardware design assurance level that is being implemented.

Appendix B contains guidance on design assurance techniques for hardware used in implementing Level A and B functions which should be applied in addition to the guidance in Section 2 through Section 11. Appendix B may be applied for hardware of design assurance Levels C and D at the applicant's discretion.

The Glossary of Terms as used in this document is contained in Appendix C. Appendix D contains a list of acronyms that are used in the document and spells out their complete names.

A list does not imply that its elements are in any way complete or that all elements are relevant to any specific product.

Notes are used in this document to provide explanatory material, emphasize a point, or draw attention to related subjects, which are not entirely within context. Notes do not contain guidance.

The word "should" is used when the intention is to provide guidance. "May" is used in conjunction with optional text.

This document uses the term "hardware item" to describe the electronic hardware which is the subject of the document.

The qualifier "hardware" is to be assumed throughout the document unless specifically stated otherwise. When the term "requirements" is used it is assumed to mean "hardware requirements". A system or software qualifier will always be specifically stated, such as "system requirement".

Note: *Various industry advisory documents and aviation requirement documents do not always use harmonized terminology. For example, Federal Aviation Regulations (FAR) 21 and Joint Aviation Requirements (JAR) 21 use the term* *"product" to mean an aircraft, aircraft engine, or propeller. Document SAE ARP4754/EUROCAE ED-79 uses the term "product" to mean hardware, software, item or system generated in response to a defined set of requirements. The reader is advised to be aware of these and other differences in the use of terminology. This document uses the definitions in the glossary.*

1.6 Complexity Considerations

Although various classifications of the term "complexity" are used to describe electronics, such as simple, complex and highly complex, the differentiation between these classifications is not rigorously defined. Defining differences in complexity herein is based on the feasibility and level of difficulty necessary to accomplish acceptable verification coverage by deterministic means.

Hardware should be examined hierarchically at the levels of integrated circuit, board and LRU for complexity, including addressing functions that may not be testable, such as unused modes in multiple usage devices and potentially hidden states in sequential machines.

A hardware item is identified as simple only if a comprehensive combination of deterministic tests and analyses appropriate to the design assurance level can ensure correct functional performance under all foreseeable operating conditions with no anomalous behavior.

When an item cannot be classified as simple, it should be classified as complex. An item constructed entirely from simple items may itself be complex. Items that contain a device, such as an ASIC or a PLD, can be considered simple if they meet the criteria of simple as described in this section.

For complex items, the proposed means of providing design assurance should be agreed to by the certification authority early in the hardware design life cycle to mitigate program risk.

For a simple hardware item, extensive documentation of the design process is unnecessary. The supporting processes of verification and configuration management need to be performed and documented for a simple hardware item, but extensive documentation is not needed. Thus, there is reduced overhead in designing a simple hardware item to comply with this document. The main impact of this document is intended to be on the design of complex hardware items.

1.7 Alternative Methods or Processes

Methods or processes other than those described in this document may be used to provide hardware design assurance. These methods and processes should be assessed based on their ability to satisfy the applicable regulations. Alternative methods or processes should be approved by the certification authority prior to their implementation. In lieu of direct comparison with the applicable regulations, the applicant could use the following guidance to reduce program risk while evaluating alternative methods or processes by comparison to this document.

Considerations for evaluation of alternative methods or processes may include:

1. Where used instead of processes prescribed by this document, processes satisfying one or more of the objectives of Section 2 through Section 9 should show an equivalent level of design assurance.
2. The effect of the proposed alternative methods or processes on satisfying the hardware design assurance objectives should be assessed.
3. The effect of the proposed alternative methods or processes on the life cycle data should be assessed.
4. The rationale for using the proposed alternative methods or processes should be substantiated by evidence that the methods or processes will produce the expected results.

1.8 Document Overview

Figure 1-1 is a pictorial overview of the sections in this document, and some of their relationships to each other and to other related processes. There is no intent to show data flow but rather to show which sections and external processes are related.

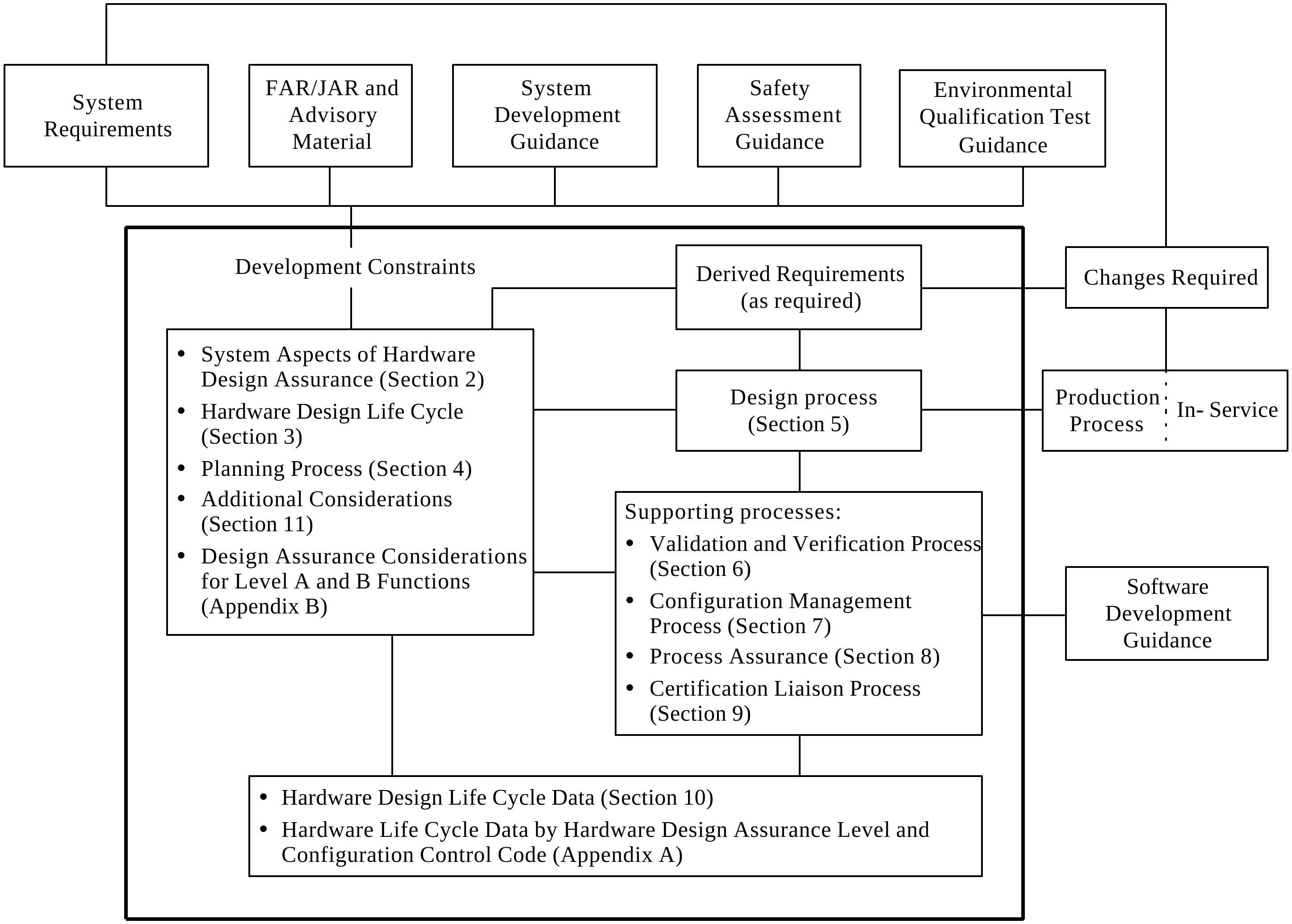

System Requirements
FAR/JAR and Advisory Material
System Development Guidance
Safety Assessment Guidance
Environmental Qualification Test Guidance
Development Constraints
Derived Requirements (as required)
Changes Required
• System Aspects of Hardware Design Assurance (Section 2)
• Hardware Design Life Cycle (Section 3)
• Planning Process (Section 4)
• Additional Considerations (Section 11)
• Design Assurance Considerations for Level A and B Functions (Appendix B)
Design process (Section 5)
Production Process
In- Service
Supporting processes:
• Validation and Verification Process (Section 6)
• Configuration Management Process (Section 7)
• Process Assurance (Section 8)
• Certification Liaison Process (Section 9)
Software Development Guidance
• Hardware Design Life Cycle Data (Section 10)
• Hardware Life Cycle Data by Hardware Design Assurance Level and Configuration Control Code (Appendix A)

Figure 1-1 Document Overview

This Page Intentionally Left Blank

2.0 SYSTEM ASPECTS OF HARDWARE DESIGN ASSURANCE

Hardware design assurance begins at the system level with the allocation of system functions to hardware and the assignment of their corresponding system development assurance levels.

A single system function may be assigned to a hardware item, to a software component or to a combination of hardware and software. Safety requirements associated with the function are addressed from a system perspective, a software perspective and a hardware perspective to determine the level of reliability and the level of assurance necessary to satisfy these requirements.

Figure 2-1 illustrates the relationships of the system development process for airborne systems and equipment and safety assessment, hardware development, and software development processes.

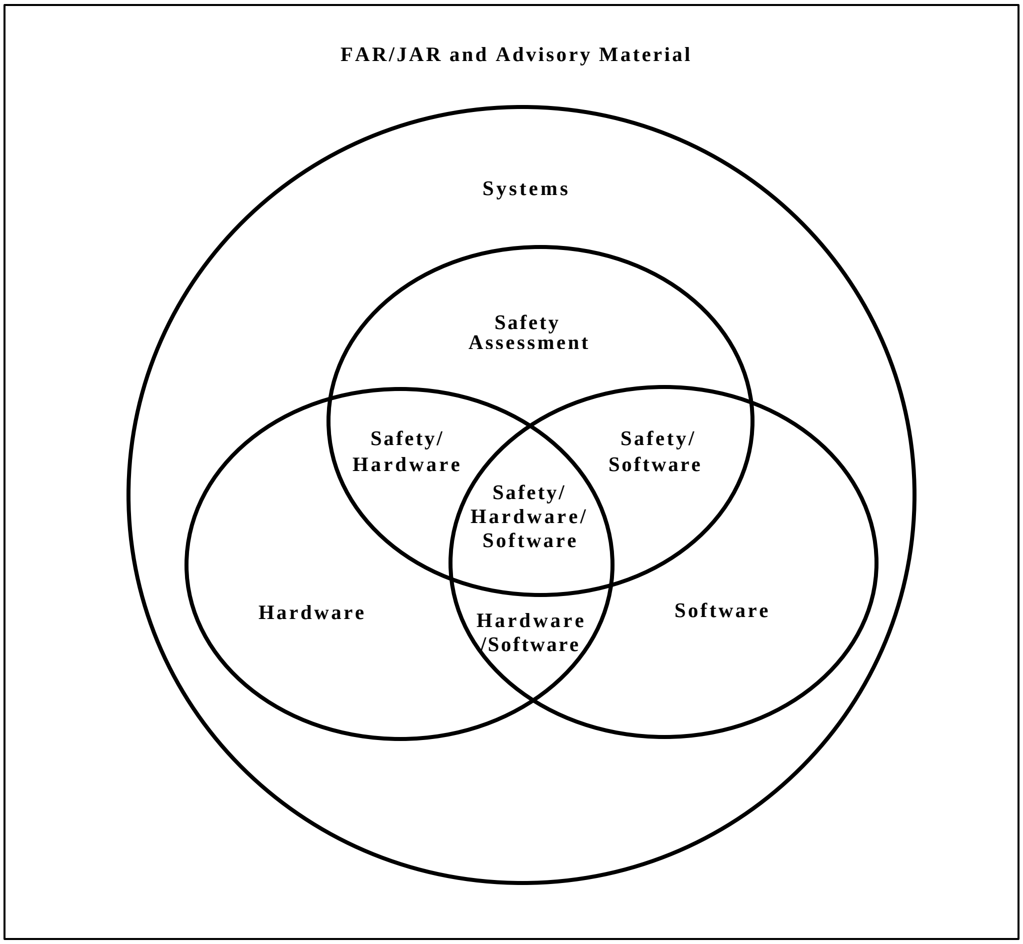


Figure 2-1 Relationships Among Airborne Systems, Safety Assessment, Hardware and Software Processes

There are four areas of overlap in the figure, Safety/Hardware, Safety/Software, Hardware/Software and Safety/Hardware/Software. These overlaps illustrate the relationship and interactions between these processes where a system requirement may result in requirements within the scope and design assurance guidance of multiple processes. For example, a hardware function that contained safety requirements would involve both the safety assessment process and the hardware design life cycle process.

The overlaps illustrate the need for a coordinated interaction between the processes to ensure that the assurance requirements of the system function are satisfied. The discussion of system or software assurance processes is beyond the scope of this document. However, in coordinating the design assurance for a hardware function, the applicant may wish to take advantage of assurance provided by activities in the systems or software processes.

These relationships and interactions are described further in Section 2.1.1 through Section 2.1.3.

2.1 Information Flow

The flow of information between the life cycle processes is shown in Figure 2-2. The following sections describe the flow of information from the system development process to the hardware design life cycle process, from the hardware design life cycle process to the system development process, and between the hardware design life cycle process and the software life cycle process.

Note: *It is recognized that these are iterative processes and changes will occur throughout the hardware design life cycle.*

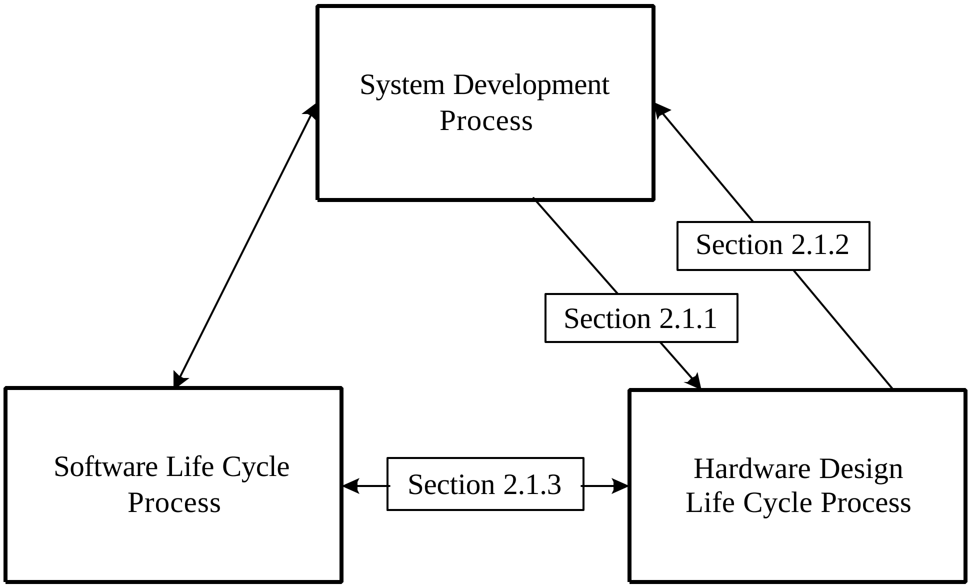

System Development Process
Section 2.1.2
Section 2.1.1
Software Life Cycle Process
Section 2.1.3
Hardware Design Life Cycle Process

Figure 2-2 System Development Processes

2.1.1 Information Flow from System Development Process to Hardware Design Life Cycle Process

This information flow may include:

1. Design and safety requirements allocated to hardware.
2. Design assurance level for each function, along with its associated requirements and failure conditions, if applicable.
3. Allocated probabilities and at risk exposure times for hardware functional failures.
4. Hardware/software interface description.
5. Requirements for safety strategies and design constraints, such as testability, design methods, and hardware architectures.
6. Requirements for system verification activities to be performed by hardware level verification.
7. Installation, ergonomic and environmental requirements allocated to hardware.
8. Integration problem reports that may have an impact on requirements. These may arise as a result of activities, such as system verification, generation of system requirements or SSA.

2.1.2 Information Flow from Hardware Design Life Cycle Process to System Development Process

This information flow may include:

1. Implementation of the requirements, such as mechanical drawings, schematics and parts lists.
2. Hardware derived requirements that may have an impact on any allocated requirement.
3. Implementation architecture, including fault containment boundaries.
4. Evidence of any required system verification and validation activities performed during the hardware design life cycle.
5. Product safety analysis data, such as:
   a. Probabilities and failure rates for designated hardware functional failures of concern to the SSA process.
   b. Common mode fault analysis.
   c. Isolation boundaries and generic fault mitigation strategies.
   d. Latency analysis data relevant to system requirements. Examples are hardware provisions for fault monitoring, fault detection intervals and undetectable faults.

6. Requirements for hardware verification activities to be performed by system level verification.

7. Assumptions and analysis methods regarding installation requirements and environmental conditions necessary for the analyses to be valid.

8. Problem or change reports that may have an impact on system, software or allocated hardware requirements.

2.1.3 Information Flow between Hardware Design Life Cycle Process and Software Life Cycle Process

This information flow may include:

1. Derived requirements needed for hardware/software integration, such as definition of protocols, timing constraints, and addressing schemes for the interface between hardware and software.

2. Instances where hardware and software verification activities require coordination.

3. Identified incompatibilities between the hardware and the software, which may be part of a reporting and corrective action system.

4. Safety assessment data that should also be made available to system processes.

2.2 System Safety Assessment Processes

There are three system safety assessment processes: functional hazard assessment (FHA), preliminary system safety assessment (PSSA) and SSA. These processes are used to establish the system safety objectives applicable to the system development assurance process, and to determine that the system functions achieve the safety objectives.

The SSA process should transform the safety objectives into system and equipment safety requirements. These requirements should embody the basic safety objectives and safety attributes for system and equipment functions and architecture. The SSA process and the system development process allocate these safety requirements to the hardware.

There are five system development assurance levels, Level A through Level E, corresponding to the five classes of failure conditions: catastrophic, hazardous/severe-major, major, minor and no effect. Table 2-1 correlates the hardware design assurance levels to the five classes of failure conditions and provides definitions of hardware failure conditions and their respective design assurance levels. Initially, the hardware design assurance level for each hardware function is determined by the SSA process using an FHA to identify potential hazards and then the PSSA process allocates the safety requirements and associated failure conditions to the function implemented in the hardware.

Throughout the hardware design life cycle, there may be iterative feedback between the safety, system and hardware processes to ensure that the hardware as designed and built will satisfy the system safety, functional and performance requirements allocated to the hardware.

Table 2-1 Hardware Design Assurance Level Definitions and their Relationships to Systems Development Assurance Level

System Development Assurance Level	Failure Condition Classification	Failure Condition Description	Hardware Design Assurance Level Definitions
Level A:	Catastrophic	Failure conditions that would prevent continued safe flight and landing.	A: Hardware functions whose failure or anomalous behavior, as shown by the hardware safety assessment, would cause a failure of system function resulting in a catastrophic failure condition for the aircraft.
Level B:	Hazardous / Severe-Major	Failure conditions that would reduce the capability of the aircraft or the ability of the flight crew to cope with adverse operating conditions to the extent that there would be: a large reduction in safety margins or functional capabilities, physical distress or higher workload such that the flight crew could not be relied on to perform their tasks accurately or completely, or adverse effects on occupants including serious or potentially fatal injuries to a small number of those occupants.	B: Hardware functions whose failure or anomalous behavior, as shown by the hardware safety assessment, would cause a failure of system function resulting in a hazardous/severe-major failure condition for the aircraft.
Level C:	Major	Failure conditions that would reduce the capability of the aircraft or the ability of the flight crew to cope with adverse operating conditions to the extent that there would be: a significant reduction in safety margins or functional capabilities, a significant increase in flight crew workload or in conditions impairing flight crew efficiency, or discomfort to occupants, possibly including injuries.	C: Hardware functions whose failure or anomalous behavior, as shown by the hardware safety assessment, would cause a failure of system function resulting in a major failure condition for the aircraft.
Level D:	Minor	Failure conditions that would not significantly reduce aircraft safety, and which would involve flight crew actions that are well within their capabilities. Minor failure conditions may include: a slight reduction in safety margins or functional capabilities, a slight increase in flight crew workload, such as routine flight plan changes, or some inconvenience to occupants.	D: Hardware functions whose failure or anomalous behavior, as shown by the hardware safety assessment, would cause a failure of system function resulting in a minor failure condition for the aircraft.
Level E:	No Effect	Failure conditions that do not affect the operational capability of the aircraft or increase flight crew workload.	E: Hardware functions whose failure or anomalous behavior, as shown by the hardware safety assessment, would cause a failure of a system function with no effect on aircraft operational capability or flight crew workload. For a function determined to be Level E, no further guidance of this document need apply, however, it may be used for reference.

2.3 Hardware Safety Assessment

The hardware safety assessment is conducted in conjunction with and to support the SSA process. The intent of this safety process is to demonstrate that the applicable systems and equipment, including the hardware, have satisfied the safety requirements of applicable aircraft certification requirements.

Given the safety, functional and performance requirements allocated to the hardware by the system process, the hardware safety assessment determines the hardware design assurance level for each function and contributes to determining the appropriate design assurance strategies to be used.

2.3.1 Hardware Safety Assessment Considerations

The designer of a hardware item may show compliance with the safety requirements allocated to the hardware and with the hardware design assurance level by an appropriate design assurance strategy.

A single design assurance level and strategy may be applied to an entire hardware item or a hardware item may be evaluated as having separate functional failure paths (FFPs) in order to accommodate a mix of design assurance levels or design assurance strategies. A functional failure path analysis (FFPA) may be used to justify a lower design assurance level for a portion of the hardware item, or to accommodate different functions implemented with different technologies or product service histories.

Note: *FFPA is described in Appendix B, Section 2. Although written to address the subject matter of Appendix B, this analysis method may be applied to any design assurance level.*

If a hardware item contains functions that individually have different design assurance levels, such situations may be addressed by either of the following methods:

- The entire item may be assured at the highest design assurance level.
- The individual hardware functions may be assured separately at their respective hardware design assurance levels as defined by the hardware safety assessment, if their function, interfaces and shared resources can be protected from adverse effects of functions of lower design assurance levels. Design assurance of shared resources should be the design assurance level of the function with the highest level.

Guidance for hardware safety assessment includes:

1. Iterative hardware safety assessment and design should determine derived hardware safety requirements and ensure that system safety requirements allocated to the hardware are satisfied and ensure that derived requirements are satisfied.
2. These derived requirements should include safety requirements for hardware architecture, circuits and components, and protection against anomalous behaviors,

including incorporating specific hardware architectural and functional safety attributes, such as:

a. Circuit or component redundancy.

b. Separation or electrical isolation between circuits or components.

c. Dissimilarity between circuits or components.

d. Monitoring of circuits or components.

e. Protection or reconfiguration mechanisms.

f. Allowed failure rates and probabilities for circuit and component random failures and latent failures.

g. Limitations of usage or installation.

h. Prevention and management of upsets and upset recovery.

3. The hardware design assurance process and the hardware safety assessment should jointly determine the specific means of compliance and design assurance level for each function and should determine that an acceptable level of design assurance has been achieved.

Note: *Anomalous behavior of the hardware may be caused by random faults or design errors in a hardware item, or by upsets to the hardware.*

The hardware designer may choose a higher hardware design assurance level for a hardware item function. An example would be the anticipation of re-using a hardware item function in an installation requiring a higher level of design assurance.

The hardware safety assessment may use various qualitative and quantitative assessment methods. These may include fault tree analysis (FTA), common mode analysis, failure modes and effects analysis, and statistical reliability analysis methods for applicable quantitative assessment of random faults.

2.3.2 Quantitative Assessment of Random Hardware Faults

Statistical failure assessment and prediction methods, which are based on hardware failure rates, redundancy, separation and isolation, failure mode statistics, probability analysis, component de-rating, stress analysis, and manufacturing process control, have proven to be acceptable means of assessing quantitative risk factors for random failures of hardware.

2.3.3 Qualitative Assessment of Hardware Design Errors and Upsets

Unlike random failures of hardware, neither design errors nor some types of upsets are statistically predictable, and both may cross redundancy boundaries in the form of common mode faults. Redundancy management techniques and quantitative assessment methods to be used should be selected so that potential common mode faults and the effects of upsets are precluded or mitigated when necessary.

Although difficult to assess quantitatively, safety risk from design errors and upsets can be effectively assessed by a practical application of qualitative safety assessment methods. Analysis techniques, such as fault tree analysis, common mode analysis, and functional failure modes and effects analysis (F-FMEA), are fundamentally qualitative methods, and can be used to address hardware design errors and upsets. More specifically, these methods can determine the potential effects of design errors and upsets, and can help determine the means by which they are to be precluded or mitigated. Using these methods, the hardware safety assessment can contribute to determining the hardware design assurance strategies to be used and can be used iteratively throughout the hardware design process to qualitatively determine the assurance achieved by the selected strategies.

2.3.4 Design Assurance Considerations for Hardware Failure Condition Classification

As the severity of the system failure condition increases, the amount of hardware design assurance necessary to ensure that related failure conditions have been mitigated increases. For all design assurance levels, an approach or strategy should be developed to ensure an appropriate level of design assurance. Figure 2-3 outlines the decision-making process for developing an appropriate design assurance strategy.

Guidance includes:

1. For Level A or B functions implemented in hardware, the design assurance considerations should address potential anomalous behaviors and potential design errors of the hardware functions.

2. The decision making process outlined in Figure 2-3 should be used when developing design assurance strategies for each hardware function being implemented.

3. The strategies described in Appendix B should be applied for Level A and B functions in addition to the guidance provided in Section 3 through Section 11.

4. The design assurance strategy should be selected as a function of the hardware architecture and usage, and of the hardware implementation technology that has been chosen.

Different technologies, components selection, and components usage offer varying degrees of hardware design life cycle information and varying degrees of inherent protection against design errors and their effects. The most suitable design assurance method may vary for different functional paths within the same hardware item.

The numbers in the decision and activity blocks of Figure 2-3 refer to the numbered items following the figure that provide further clarification of the decision or activity.

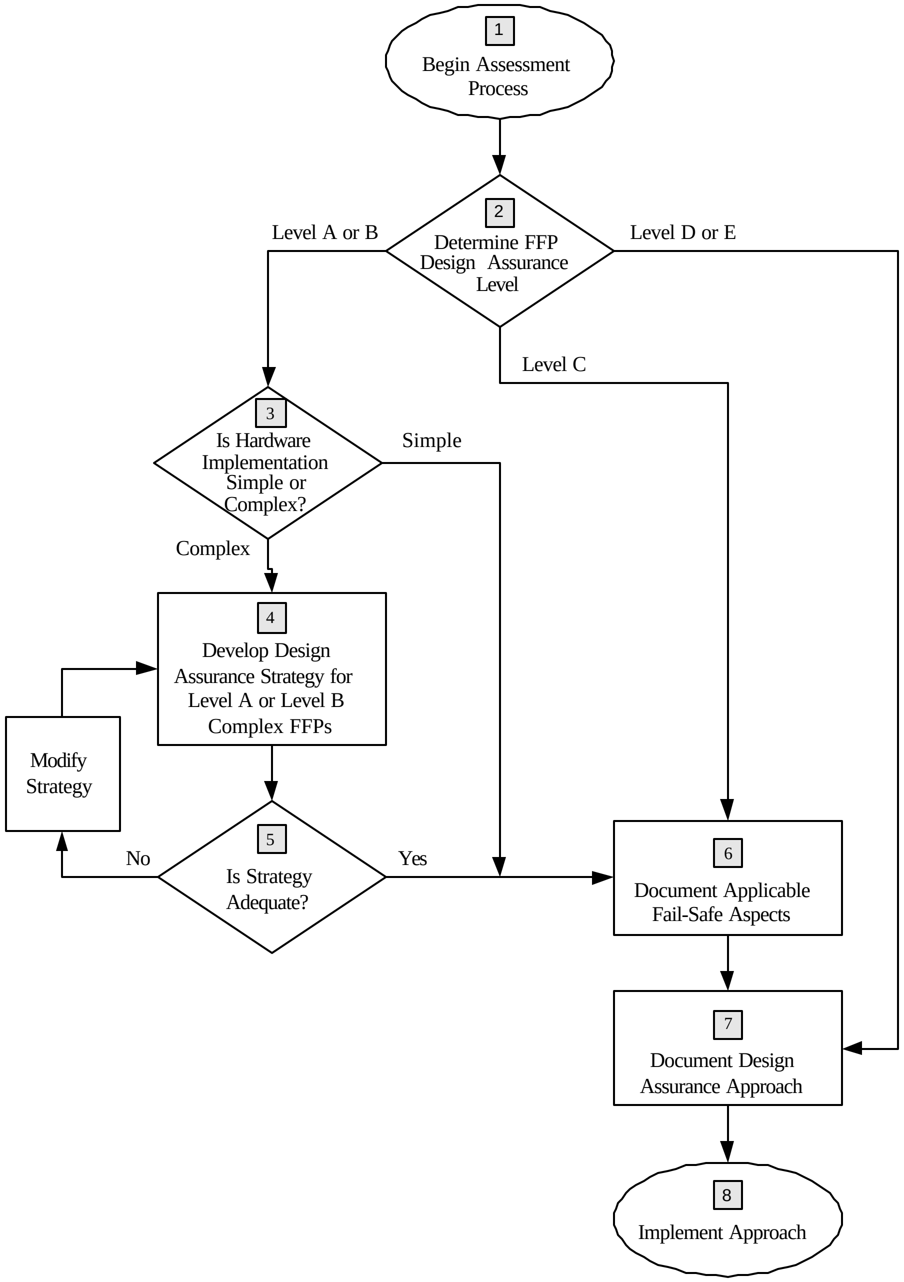

1
Begin Assessment Process
2
Determine FFP Design Assurance Level
Level A or B
Level D or E
Level C
3
Is Hardware Implementation Simple or Complex?
Simple
Complex
4
Develop Design Assurance Strategy for Level A or Level B Complex FFPs
Modify Strategy
5
Is Strategy Adequate?
No
Yes
6
Document Applicable Fail-Safe Aspects
7
Document Design Assurance Approach
8
Implement Approach

Figure 2-3 Decision Making Process for Selecting the Hardware Design Assurance Strategy

1. **Begin Assessment Process.** For all design assurance levels, an approach or strategy should be developed to ensure an appropriate level of design assurance.

2. **Determine FFP Design Assurance Level.** For each identified hardware item, determine and document the FFPs associated with the item and the design assurance level. Conventional safety assessment techniques should be used to determine which hardware circuits are and which are not in the identified Level A or B FFPs.

3. **Is the Hardware Implementation of the FFP Simple or Complex?** For hardware design assurance Level A or Level B FFPs, determine if the hardware is simple or complex as described in Section 1.6.

4. **Develop Design Assurance Strategy for Level A or Level B Complex FFPs.** If the FFP is complex and Level A or B, use the additional strategies described in Appendix B to determine the appropriate design assurance strategy, corresponding implementation concept and the error mitigation methods. For each Level A or B FFP, a design assurance strategy should be determined using advanced analysis, product service experience or architectural mitigation.

   Level A FFPs in an implementation may require more than one approach if the approach selected does not provide complete mitigation of potential failures and anomalous behaviors.

5. **Is the Strategy Adequate?** Determine if there are deficiencies in the design assurance strategies and, if deficiencies exist in the strategy or would exist in the data expected to be available, modify the strategy to correct the deficiencies by proposing additional design assurance, implementation or architectural strategies.

   When the design assurance strategy is acceptable, document the design assurance processes for each FFP. The strategy should also address certification authority participation aspects, such as schedule milestones, program reviews and oversight activities.

6. **Document the Applicable Fail-Safe Aspects.** Determine the appropriate fail-safe design architecture and features of the hardware item and perform an analysis to satisfy the availability and integrity requirements of the system. Document the fail-safe design aspects and the associated common mode analysis, probability analysis, architecture and other features.

7. **Document the Design Assurance Approach and Strategy.** Document and obtain certification authority approval of the applicable design assurance approach and strategy in the system certification plan or the Plan for Hardware Aspects of Certification (PHAC).

8. **Implement the Approach.** Implement the hardware design in compliance with the appropriate design assurance approach as defined in the approved plan and document evidence of compliance to approved plans and strategy.

3.0 HARDWARE DESIGN LIFE CYCLE

This section outlines the hardware design life cycle discussed in Section 4 through Section 9. This document does not prescribe a preferred life cycle model, nor imply a structure for the performing organization. The hardware design life cycle is equally applicable to the development of new systems or equipment and modifications to existing systems or equipment. The life cycle for each project should be based on selection and arrangement of processes and activities determined by the attributes of the project, such as requirements stability, use of previously developed hardware and hardware design assurance levels. The hardware design life cycle processes may be iterative, that is, entered, re-entered and modified due to incremental development and feedback between the processes.

3.1 Hardware Design Life Cycle Processes

The hardware design life cycle processes are:

1. The hardware planning process, described in Section 4, defines and coordinates the activities of the hardware design and supporting processes for a project.
2. The hardware design processes, described in Section 5, generate the design data and resultant hardware item. These processes are requirements capture, conceptual design, detailed design, implementation and production transition.
3. The supporting processes, described in Section 6 through Section 9 produce the hardware design life cycle data that assures correctness and control of the hardware design life cycle and its outputs, including planning, design, hardware safety assessment and supporting processes. These processes are typically performed concurrently with the planning and design processes. These processes are validation, verification, configuration management, process assurance and certification liaison.

3.2 Transition Criteria

The challenges of developing a hardware item with different subitems at different stages of development require a means to provide a reasonable amount of control of the design process in order to manage the risk of starting the next process before all elements of the previous process are complete. Transition criteria, defined as the minimum data used to assess movement from one process to another, may be used at key process points. Analysis during the planning process should determine the use of transition criteria. It is not necessary to establish transition criteria between each pair of process steps defined in the plans. The selection of transition criteria should address the impact on safety. For example, before performing verification of a function for certification credit, the requirements for that function need to be documented and the implementation of that function needs to be under configuration management.

Transition criteria should be documented in the hardware plans. Use of transition criteria does not imply any particular life cycle model or prevent such development strategies as rapid prototyping and concurrent engineering.

This Page Intentionally Left Blank

4.0 PLANNING PROCESS

This section describes the hardware planning process used to control the development of the hardware item. This process produces the hardware plans, which may be contained in one or more documents. If multiple documents are used, the main plan should contain appropriate references to the supporting documents. Standard documents covering specific hardware design life cycle processes, such as configuration management or process assurance, are acceptable provided they meet the planning objectives for the applicable process.

4.1 Planning Process Objectives

The purpose of the hardware planning process is to define the means by which the functional and airworthiness requirements are converted into a hardware item with an acceptable amount of evidence of assurance that the item will safely perform its intended functions. The objectives of the hardware planning process are:

1. The hardware design life cycle processes are defined.

   Note: *Activities, milestones, inputs, outputs and organizational responsibilities may be included in the plans.*

2. Standards are selected and defined.

3. The hardware development and verification environments are selected or defined.

4. The means of compliance of the hardware design assurance objectives, including strategies identified using guidance in Section 2.3.4, are proposed to the certification authority.

Note: *New and evolving technologies, tools and processes may require details of the planning process to change. Therefore, flexibility is a key element of the planning process.*

4.2 Planning Process Activities

Guidance for the planning process includes:

1. The hardware design life cycle process, including transition criteria, if applicable, and the inter-relationships between the individual processes, such as their sequencing and feedback mechanisms, should be defined.

2. The proposed design methods should be defined and explained. This includes consideration of the expected hardware design and the rationale of the proposed verification methods.

3. Hardware design standards, if any are to be used for the project, including acceptable deviations from the standards, should be identified. These may range from generic quality standards to company or program specific standards.

Note: *Standards help reduce the probability of undetected design errors by providing a compilation of proven engineering practices determined from past developments.*

The applicant and hardware developer should be aware when applying standards to new designs and new technologies, that the applicability may be invalid. Deviations from these standards may be necessary due to design constraints, conflicts with system requirements or incompatibility with new technologies. The planning process is an opportunity to review what deviations may be acceptable if standards are used.

4. The means of achieving coordination between the hardware design processes and the supporting processes, with particular attention to activities associated with systems, software and aircraft certification, should be determined.

   Note: *Coordination may be in the form of a schedule showing milestones for events to accomplish the objectives of the processes described in this document.*

5. The activities of each hardware design process and associated supporting processes should be defined. The definition should be at a level that enables the hardware design process and associated supporting processes to be controlled.

6. The design environment should be chosen, including the tools, procedures, software and hardware that are to be used to develop, verify and control the hardware item and the life cycle data.

   a. If certification credit is sought for use of tools in combination, the sequence of operation of the tools should be specified in the respective plan.

   b. The design environment can affect the design of a product. Section 11.4 provides guidance for the assessment of tools and determining when tool qualification may be necessary.

7. The process for deviating from the established plans, if deviations become necessary and affect certification, should be identified.

8. The policies, procedures, standards and methods to be used to identify, manage, and control the hardware, the associated baselines, and the hardware design life cycle data should be described.

9. Where the applicant intends to use subcontractors for all or part of the hardware design life cycle, the hardware plans should identify the method for ensuring that the design assurance objectives are met.

10. The policies and procedures for implementation of process assurance of the hardware design processes should be described.

11. Verification process independence, process assurance independence and associated organizational responsibilities should be described in the PHAC.

12. The means to satisfy the objectives of this guidance should be recorded and communicated to the certification authority early in the process. These means should be recorded in the PHAC.

    <u>Note</u> : *Timely coordination of any changes to these means is encouraged to maximize acceptance of the resultant certification data as proper evidence of meeting the design assurance requirements.*

This Page Intentionally Left Blank

5.0 HARDWARE DESIGN PROCESSES

The hardware design processes produce a hardware item that fulfills the requirements allocated to hardware from the system requirements. This section describes five major processes as depicted in Figure 5-1. These are Requirements Capture, Conceptual Design, Detailed Design, Implementation and Production Transition. These design processes may be applied at any hierarchical level of the hardware item, such as LRUs, circuit board assemblies and ASICs/PLDs. The following sections describe each process, its objectives and the related activities that should be addressed to reduce the probability of design and implementation errors that affect safety. It is important that each of these processes is planned and the details recorded in a hardware design plan.

Each process, and interactions between the processes, can be iterative. For each iteration, the effect of the change on each of the processes should be addressed and evaluated for impact on the results of previous iterations.

Note 1: *It is good engineering practice to document process artifacts, such as design notes, design review notes and problem reports, throughout the design process.*

Current practices provide many different means, graphical, mathematical, database or text based, to represent requirements and design implementations. Examples of these representations are schematics, hardware description languages (HDL), state diagrams, Boolean representations and graphical methods.

Note 2: *Some representations are adapted to a specific process or combination of processes, such as requirements capture, conceptual design or detailed design, and some are adapted to more efficiently implement a specific implementation technology. Evidence to support the design assurance level should be provided, regardless of the design representation used.*

For each design representation used, the following should be considered:

1. The guidance of this document should be followed regardless of the representation, or combination of representations, used.

2. The design representation should allow the hardware item to be consistently replicated.

3. Small changes in design representation may have a large impact on the design implementation. The impact of these changes on design assurance should be addressed.

4. The design representation environment or method may change after the baseline of the design data has been established. If this occurs, the impact of the change on the replication of the design should be assessed.

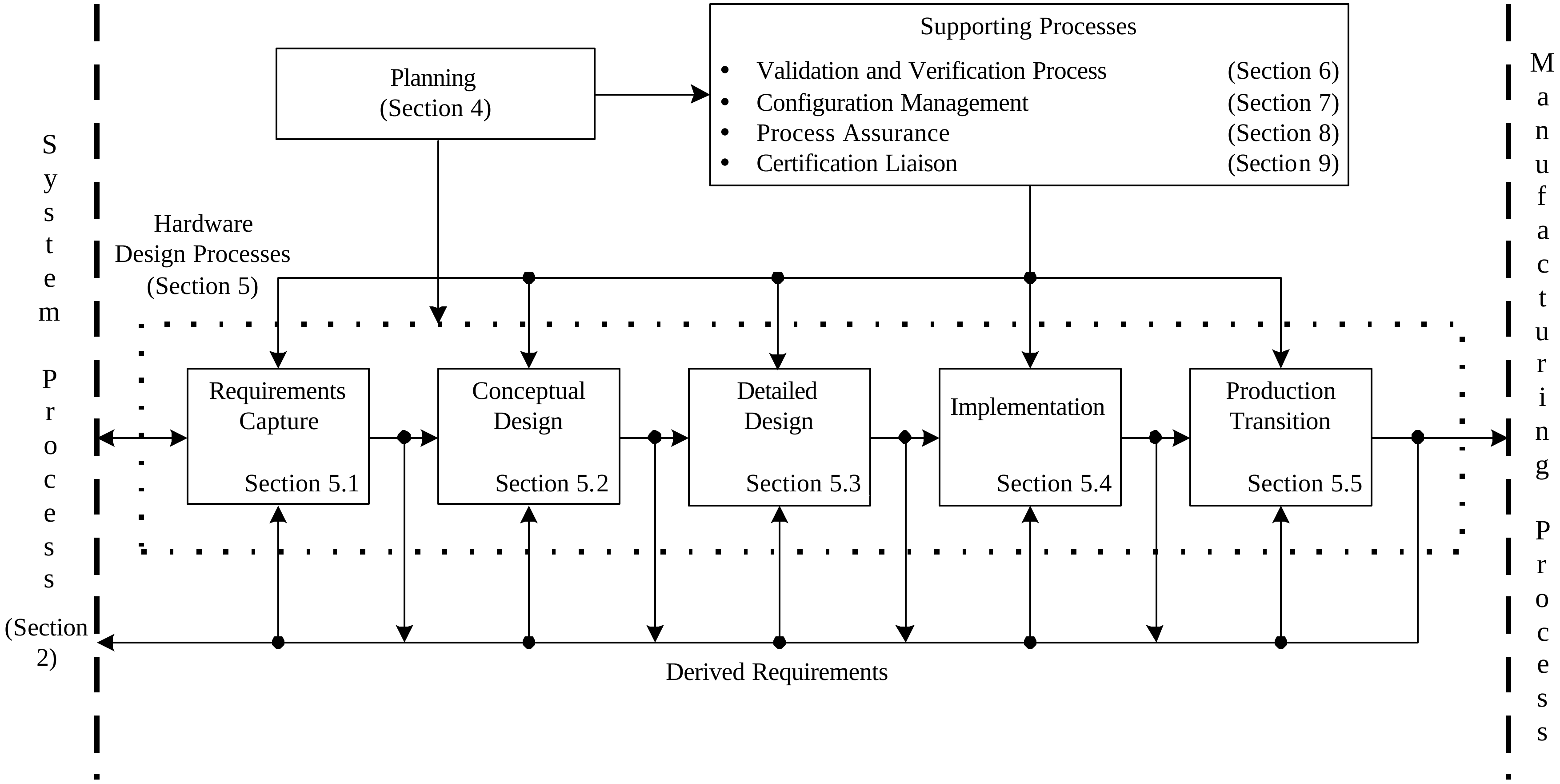

Supporting Processes
• Validation and Verification Process (Section 6)
• Configuration Management (Section 7)
• Process Assurance (Section 8)
• Certification Liaison (Section 9)
Planning (Section 4)
Hardware Design Processes (Section 5)
System Process (Section 2)
Manufacturing Process
Requirements Capture Section 5.1
Conceptual Design Section 5.2
Detailed Design Section 5.3
Implementation Section 5.4
Production Transition Section 5.5
Derived Requirements

Figure 5-1 Hardware Design Life Cycle

HDL design representations use coded text based techniques that are similar in appearance to those used for software representations. This similarity in appearance can mislead one to attempt to use software verification methods directly on the design representation of HDL or other equivalent hardware specification languages. The guidance of this document is applicable for design assurance for designs using an HDL representation.

Note: *The structured processes described throughout this document are applicable to complex hardware designs including ASICs and PLDs. As an example, the following table maps typical ASIC/PLD processes to the processes depicted in Figure 5-1 of this document.*

***Table 5-1* Typical ASIC/PLD Process Mapping**

Typical ASIC/PLD Process	***Process***
Part of higher level planning	*Planning (Section 4)*
ASIC/PLD Architectural Decisions	*Safety Assessment (Section 2.3)*
ASIC/PLD Requirements Capture	*Requirements Capture (Section 5.1)*
ASIC/PLD Preliminary Design including behavioral design	*Conceptual Design (Section 5.2)*
ASIC/PLD Detailed Design including synthesis, mask generation and fuse file	*Detailed Design (Section 5.3)*
ASIC/PLD Fabrication including external fabrication and test as well as programming programmable components	*Implementation (Section 5.4)*
ASIC/PLD Production Transition	*Production Transition (Section 5.5)*
ASIC/PLD Validation and Verification including timing analysis, behavioral simulation, gate level simulation and design	*Validation and Verification Process (Section 6)*
ASIC/PLD Configuration Management including tools and part database	*Configuration Management Process (Section 7)*

5.1 Requirements Capture Process

The requirements capture process identifies and records the hardware item requirements. This includes those derived requirements imposed by the proposed hardware item architecture, choice of technology, the basic and optional functionality, environmental, and performance requirements as well as the requirements imposed by the system safety assessment. This process may be iterative since additional requirements may become known during design.

5.1.1 Requirements Capture Objectives

The objectives for the requirements capture process are:

1. Requirements are identified, defined and documented. This includes allocated requirements from the PSSA and derived requirements from the hardware safety assessment.

   Note: *Traceability of verification results to the hardware requirements is addressed in Section 6. It is desirable to establish this method of traceability during the requirement capture process.*

2. Derived requirements produced are fed back to the appropriate process.

3. Requirement omissions and errors are provided to the appropriate process for resolution.

5.1.2 Requirements Capture Activities

The requirements capture activities form an iterative process which helps assure consistency of the requirements with the design implementation, the system requirements and the software requirements.

Guidance for the requirements capture activities includes:

1. The system requirements allocated to the hardware item should be documented. These may include identifying requirements, such as functionality and performance, and architectural considerations, such as segregation, Built-In-Test, testability, external interfaces, environment, test and maintenance considerations, power, and physical characteristics.

2. The safety requirements from the PSSA related to the hardware item should be identified. These may include:

   a. Design assurance levels imposed on the functions to be implemented in the hardware.

   b. Probabilistic requirements for malfunctions or loss of function.

   c. Hardware architectural and functional safety attributes, such as those outlined in Section 2.3.1, selected to meet the functional allocation.

3. Design constraints due to production processes, standards, procedures, technology, design environment and design guidance should be identified.

4. Derived requirements necessary for implementation should be determined. Requirements derived from the hardware safety assessment that have safety implications should be uniquely identified.

   Note: *Derived requirements may address conditions, such as:*

   a. Specific constraints to ensure that functions of a higher design assurance level can withstand anomalies of functions of a lower *design assurance level as seen at the interface of the function with the lower design assurance level.*

b. *The range of data inputs considering typical and full-scale data values as well as the high and low states of bits in data words or control registers.*

c. *Power-up reset or other reset states.*

d. *Supply voltage and current demands.*

e. *Performance of time-related functions, such as filters, integrators and delays.*

f. *State machine transitions that are possible, whether they are anticipated or not.*

g. *Signal timing relationships or electrical conditions under normal and worst-case conditions.*

h. *Signal noise and cross-talk.*

i. *Signal glitches in asynchronous logic circuits.*

j. *Specific constraints to control unused functions.*

5. Derived requirements should be fed back to the SSA process so that the effects on the system requirements can be assessed.

6. The requirement data should be documented in quantitative terms, with tolerances where applicable. This does not include the description of design or verification solutions.

7. Requirement omissions or errors discovered during this process should be provided to the system development process.

8. The requirements, including those generated to meet the PSSA requirements, should be traceable to the next higher hierarchical level of requirements. Derived requirements should be identified and traced as far as possible through the hierarchical levels.

Note: *System level validation of allocated hardware safety requirements may occur during the requirement capture process. Validation of derived hardware requirements is described in Section 6.1.*

5.2 Conceptual Design Process

The conceptual design process produces a high-level design concept that may be assessed to determine the potential for the resulting design implementation to meet the requirements. This may be accomplished using such items as functional block diagrams, design and architecture descriptions, circuit card assembly outlines, and chassis sketches.

5.2.1 Conceptual Design Objectives

The conceptual design objectives are:

1. The hardware item conceptual design is developed consistent with its requirements.
2. Derived requirements produced are fed back to the requirements capture or other appropriate processes.
3. Requirement omissions and errors are provided to the appropriate processes for resolution.

5.2.2 Conceptual Design Activities

Guidance for the conceptual design activities includes:

1. A high-level description should be generated for the hardware item. This may include:
   a. Architectural constraints related to safety, including those necessary to address design errors and functional, component over-stress, reliability and robustness defects.
   b. Identification of any implementation constraints on software or other system components.
2. Major components should be identified. The way they contribute to the hardware safety requirements should be determined, including the impact of unused functions.
3. Derived requirements, including the interface definition, should be fed back to the requirements capture process.
4. Requirement omissions and errors should be fed back to the appropriate process for resolution.
5. The reliability, maintenance and test features to be provided should be identified.

<u>Note:</u> *Consensus between the relevant parties that the conceptual design objectives have been met is recommended. Typically, a design review is used to accomplish this consensus.*

5.3 Detailed Design Process

The detailed design process produces detailed design data using the hardware item requirements and conceptual design data as the basis for the detailed design.

5.3.1 Detailed Design Objectives

The detailed design process objectives are:

1. The detailed design is developed from the hardware item requirements and conceptual design data.

2. Derived requirements are fed back to the conceptual design process or other appropriate processes.

3. Requirement omissions or errors are provided to the appropriate processes for resolution.

5.3.2 Detailed Design Process Activities

Guidance for the detailed design activities includes:

1. The detailed design data for the hardware item should be generated based on the requirements and conceptual design data. This may include assembly and interconnection data, component data, HDL, test methods and hardware-software interface data.

   Note: *During the detailed design process, verification methods are used informally to facilitate the technical decisions made during this process. For example, analysis of design parameters, such as logic timing and parameter variations, can provide information on which to base design decisions.*

2. Architectural design techniques should be implemented as necessary. These may include establishing safety monitors for proper functionality, dissimilarity between function and safety monitors, preclusion of a design error from impacting safety, and fault tolerant designs.

3. Test features should be designed in, where necessary, to allow verification of safety requirements.

   Note: *It is important to develop the design in a way that certain safety features can be verified not only during the hardware design life cycle, but also as a part of an acceptance test and a field return to service test.*

4. An assessment of unused functions should be performed to identify potential effects on safety. Adverse effects should be addressed.

5. Constraints on the design, installation or operation of the hardware item that, if not adhered to, could affect the safety of the item should be identified.

6. Derived requirements produced during the detailed design process should be fed back to the conceptual design or other appropriate processes.

7. Requirement omissions and errors discovered during the detailed design process should be provided to the appropriate process for resolution.

5.4 Implementation Process

The implementation process uses the detailed design data to produce the hardware item that is an input to the testing activity.

5.4.1 Implementation Objectives

The objectives of the implementation process are:

1. A hardware item is produced which implements the hardware detailed design using representative manufacturing processes.
2. The hardware item implementation, assembly and installation data is complete.
3. Derived requirements are fed back to the detailed design process or other appropriate processes.
4. Requirement omissions and errors are provided to the appropriate processes for resolution.

5.4.2 Implementation Activities

Guidance for the implementation activities includes:

1. A hardware item should be produced using the design data and, where practical, the resources intended for the production product. This may include procurement, kitting, build, inspection and test.
2. Derived requirements generated by the implementation process should be fed back to the detailed design process or other appropriate processes.
3. Omissions and errors discovered during the implementation process should be provided to the appropriate process for resolution.

5.5 Production Transition Process

In this process, manufacturing data, test facilities and general resources should be examined to ensure availability and suitability for production. The production transition process uses the outputs from the implementation and verification processes to move the product into production.

5.5.1 Production Transition Objectives

The objectives of this process are:

1. A baseline is established that includes all design and manufacturing data needed to support the consistent replication of the hardware item.
2. Manufacturing requirements related to safety are identified and documented and manufacturing controls are established.
3. Derived requirements are fed back to the implementation process or other appropriate processes.
4. Errors and omissions are provided to the appropriate processes for resolution.

5.5.2 Production Transition Activities

Guidance for the production transition activities includes:

1. Manufacturing data should be prepared from configured design data.
2. Manufacturing data should be checked for completeness and consistency with the configured design data.

   Note: *It is beyond the scope of this document to impose any conditions on the nature of the manufacturing build documentation.*

3. Any changes or improvements that are incorporated during the production transition process should be evaluated to ensure they adhere to all product requirements, especially safety requirements. Any changes not compliant with customer or certification requirements should be approved by the relevant parties.
4. Manufacturing requirements pertaining to safety should be explicitly defined so they can be controlled during the production process.
5. Data required to develop acceptance test criteria should be determined.
6. Omissions or errors that are identified should be provided to the appropriate process for resolution.

5.6 Acceptance Test

An acceptance test demonstrates that the manufactured, modified or repaired product performs in compliance with the key attributes of the unit on which certification is based. These key attributes are chosen using engineering judgement and are indicative that the product is capable of meeting the requirements to which the unit was developed.

Note 1: Configuration control of the "as built" product is not a function to be performed by the acceptance test activity. The configuration management plan, as described in Section 7 of this document, should describe how the applicant plans to perform this activity.

The scope of this document does include the determination of the acceptance test criteria, including pass/fail conditions. Production activities, including acceptance testing, are considered to be outside the scope of this document

Note 2: An acceptance test is not intended to verify all requirements on each production unit.

Subitem testing may be used as a part of the acceptance test.

Acceptance test criteria should ensure that:

1. Electrical tests are identified.
2. Environmental screening tests are identified when necessary.

3. The acceptance test provides coverage of those design aspects necessary to meet the safety requirements. Safety related item or subitems that are not covered by the test should be identified and other assurance means provided. These means may include analysis, design control, statistical process control or other means as appropriate.

5.7 Series Production

This process is not within the scope of this document, but elements impacting design assurance are briefly described to complete the life cycle.

This process reproduces the hardware item on a routine basis that complies with the production data and requirements.

Considerations include:

1. Management of change of the production processes or the design provides assurance that change does not adversely impact existing safety or certification or compliance to the requirements.

   Note: In addition to the guidance proposed by the body of the document, Section 11.1.1 covers Modifications to Previously Developed Hardware. When addressing component obsolescence, refer to Section 11.2.

2. Updating of all documentation related to changes is performed in compliance with approved configuration management plans.

6.0 VALIDATION AND VERIFICATION PROCESS

This section describes the validation process and the verification process. The validation process provides assurance that the hardware item derived requirements are correct and complete with respect to system requirements allocated to the hardware item. The verification process provides assurance that the hardware item implementation meets all of the hardware requirements, including derived requirements.

6.1 Validation process

The validation process discussed here is intended to ensure that the derived requirements are correct and complete with respect to he system requirements allocated to the hardware item through the use of a combination of objective and subjective processes. Validation may be conducted before or after the hardware item is available, however, validation is typically conducted throughout the design life cycle.

<u>*Note 1:*</u> *Experience indicates that attention to the development and validation of requirements can identify subtle errors or omissions early in the development cycle and reduce exposure to subsequent redesign or inadequate hardware performance.*

The validation process discussed here is not intended to validate the requirements allocated from system requirements since validation of these requirements is assumed to occur as part of the system process. In addition, not all hardware item derived requirements need to be validated.

Design decisions that affect the system safety or functional requirements allocated to other portions of the system should be classified as derived requirements and should be validated. Additionally, design decisions and assumptions that constrain subsequent design tasks should be validated as derived requirements.

Derived requirements that need to be validated should be validated against the system requirements allocated to the hardware item. Derived requirements that are not traceable to a higher level requirement should be validated against the design decision from which they are derived.

<u>*Note 2:*</u> *A design decision to include a separate power supply for circuitry performing a specific function could result in the derivation of requirements to guide the design of that power supply. These derived requirements could include safety requirements based on the failure condition that could result from the fault or failure of the function supported by the circuit that receives power from the power supply. These requirements should be validated.*

Another example of a design decision that becomes a derived requirement is the memory address assignments for peripheral devices. There is often no requirements basis for the assignments, however, once made they constrain subsequent design tasks to comply with those assignments in order for the *design to function correctly. This derived requirement may not need to be validated.*

6.1.1 Validation Process Objectives

The objectives of the validation process for derived hardware requirements are:

1. Derived hardware requirements against which the hardware item is to be verified are correct and complete.
2. Derived requirements are evaluated for impact on safety.
3. Omissions and errors are fed back to the appropriate processes for resolution.

6.1.2 Validation Process Activities

The hardware validation objective may be satisfied through a combination of activities, such as reviews, simulation, prototyping, modeling, analysis, service experience, engineering assessment, or the development and execution of tests.

Guidance for validation process activities includes:

1. The derived hardware requirements that need to be validated should be identified.
2. For each requirement that was identified in item 1, the validation completion criteria should be identified and satisfied as shown below:

   a. Each requirement has been validated at some hierarchical level by review, analysis or test.

   b. The review, analysis or test of each requirement is appropriate for validating the requirement, especially with respect to safety.

   c. The review, analysis or test results associated with the validation of each requirement are correct and that discrepancies between actual and expected results are explained. When expected results are not pre-defined as may be the case for reviews and analyses, the results of the validation activity should be consistent with the requirement, especially with respect to safety requirements.

   Note: *Validation completion criteria may be based on requirements, safety considerations, operational mode or implementation.*

3. The derived requirements should be evaluated for their impact on safety.
4. The derived hardware requirements should be evaluated for completeness with respect to the system requirements allocated to the hardware item. For the purposes of this process, a set of requirements is complete when all the attributes that have been defined are necessary and all the necessary attributes have been defined.
5. The derived hardware requirements should be evaluated for correctness with respect to the system requirements allocated to the hardware item. For the purposes of this

document, a requirement is correct when the requirement is defined without ambiguity and there are no errors in the defined attributes.

6. Traceability between the derived hardware requirements and the validation activities and results should be established.

7. Requirement omissions and errors should be fed back to the appropriate process for resolution.

6.2 Verification Process

The verification process provides assurance that the hardware item implementation meets the requirements. Verification consists of reviews, analyses and tests applied as defined in the verification plan. The verification process should include an assessment of the results.

Note 1: *Safety aspects of hardware design take the form of safety requirements to be met by the hardware implementation.*

This section provides guidance for the verification process that should be applied to the hardware design. The verification process may be applied at any level of the design hierarchy as defined in the hardware verification plan. For safety requirements, it is advantageous to apply the verification process at various stages of the design process to increase the probability, to a high degree of confidence, that design errors have been eliminated. Some design assurance levels require that the objectives of the verification process be met with independence as addressed in Appendix A.

The software verification, software/hardware integration verification and systems integration verification processes are not addressed here. However, verification of hardware requirements during these processes is a valid method of hardware verification.

Changes to a verified configuration may be re-verified by similarity, analysis, newly designed tests or by repeating a portion of the original verification.

Note 2: *Informal testing outside the documented verification process is recommended. The procedures and results, however, are not necessarily maintained under configuration management control but are highly effective in the detection and elimination of design errors early in the design process. Verification credit can be taken for this testing only if it is formalized.*

6.2.1 Verification Process Objectives

The objectives of the verification process are:

1. Evidence is provided that the hardware implementation meets the requirements.

2. Traceability is established between hardware requirements, the implementation, and the verification procedures and results.

3. Acceptance test criteria are identified, can be implemented and are consistent with the hardware design assurance levels of the hardware functions.

4. Omissions and errors are fed back to the appropriate processes for resolution.

6.2.2 Verification Process Activities

Verification process objectives may be satisfied through a combination of methods, such as reviews, analyses, and the development and execution of tests. The verification plan documents the verification activities that should be employed to demonstrate compliance to the requirements.

Verification activities include:

1. Requirements that need a verification activity should be identified. It is not intended that requirements should be verified at every hierarchical level; requirements can be verified at a higher hierarchical level.

2. Verification methods, such as tests, simulation, prototyping, analyses and reviews, should be selected and performed.

3. Traceability between requirements, implementation, and the verification procedure and results should be established. Traceability should be consistent with the design assurance level of the function performed by the hardware. It is not intended to require traceability to detailed components, such as resistors, capacitors or gates, unless required for safety considerations.

4. Verification coverage analysis should be performed to determine that the verification process is complete, including:

    a. Each requirement has been verified at some hierarchical level by review, analysis or test.

    b. The review, analysis or test of each requirement is appropriate for verifying the requirement, especially with respect to safety requirements.

    c. The review, analysis or test results associated with the verification of each requirement are correct and that discrepancies between actual and expected results are explained. When expected results are not pre-defined as may be the case for reviews and analyses, the results of the verification activity should be consistent with the requirement, especially with respect to safety requirements.

5. The results of the verification activities should be documented.

6. Omissions and errors should be fed back to the appropriate process for resolution.

6.3 Validation and Verification Methods

This section describes some methods that may be applicable to both validation and verification.

6.3.1 Test

Test is a method that confirms that the hardware item correctly responds to a stimulus or series of stimuli. Examples of tests include functional test on the hardware item, system bench test, system validation facility test and aircraft test.

Tests may be conducted using manual, automated or specialized test equipment. Tests may also take advantage of internal hardware item test capabilities, such as Built-In-Test, in the verification process.

When it is not feasible to verify specific requirements by exercising the hardware item in its intended operational environment, other verification means should be provided, and justified.

Tests may be performed during various hardware design processes. Testing performed for certification credit requires a configured item. Systems integration or software/hardware integration test results may also be used for test credit.

Guidance for tests includes:

1. Each requirement to be validated or verified by test should be identified. Environmental qualification test requirements are part of these requirements.
2. The testing stimulus, sequence and test conditions, such as item ambient temperature and applied voltage, should be defined for each test.
3. Pass/fail criteria and a method for recording the results should be defined prior to test execution.
4. The complete identification of the test equipment and calibration date for each should be recorded.
5. The configuration identity of the hardware item being tested should be recorded.
6. Test results should be recorded and retained.
7. Test failures should be fed back to the appropriate process for resolution.

6.3.2 Analysis

Analysis is a detailed, repeatable, analytical method for evaluation of specific hardware item characteristics to demonstrate that a specific requirement is met. Examples of analyses are stress analysis, design margin analysis, common mode failure analysis, worst case analysis and test coverage analysis. Service experience may provide data for various analyses.

Note: *As the complexity of the hardware design increases, it is advantageous to make use of computerized tools, such as simulation to verify requirements and implementation of the design.*

Analyses may include a detailed examination of the functionality, performance, traceability and safety implications of a hardware item function and its relationship to other functions within the airborne system or equipment. Analysis alone or in combination with other verification methods provides evidence that a requirement is correctly implemented. Analysis should be based on data provided by the design process, service experience or other available databases.

Simulation is an important design analysis tool both for visualization of circuit operation and for higher level functional operation. Simulation can be used to analyze the impact of production variations in hardware parameters that would be difficult to do using other verification means and thus build confidence in reduction of design errors affecting safety due to these variations. Since the results depend on the models and scenarios employed, simulation results alone cannot be used for the purpose of certification credit without supporting evidence of their validity.

Examples of analysis include:

1. **Thermal Analysis.** Thermal analysis verifies that the design implementation meets the requirements when exposed to the operating thermal environment.

2. **Stress Analysis.** Stress analysis verifies that components meet de-rating criteria over the required operating range.

3. **Reliability Analysis.** Reliability analysis establishes whether the design implementation satisfies the reliability requirements of the product.

4. **Design Margin Analysis.** Design margin analysis verifies that the design implementation satisfies its functional requirements given the variability of components.

5. **Similarity Analysis.** Similarity analysis compares characteristics and usage to those of systems previously certified.

6. **Simulation Analysis.** A simulation analysis compares the simulation results and expected results.

6.3.3 Reviews

A review is a qualitative method for evaluation of the plans, requirements, design data, design concept or design implementation.

Reviews should be held throughout the hardware design life cycle as identified in the relevant plan. All reviews to be used for certification credit should be identified in the validation and verification plan.

Guidance for reviews may include:

1. Participants should have the knowledge necessary to perform the reviews.

2. Hardware review results may be used to permit or deny transitions between hardware design life cycle process activities.

3. Results of review should be documented, including decisions made and disposition of actions to be taken.

6.3.3.1 Requirements Review

The requirements review is a method to ensure the acceptability of requirements. A requirements review may address objectives from both the validation and the verification processes within the same review.

Requirement changes that occur after the initial requirements review should be subject to the same review process used initially or an equivalent review process. It is not the intent of this review to validate the system requirements allocated to the hardware item.

Guidance for requirements review includes:

1. Each requirement should be unambiguous, verifiable, and described in complete enough detail for its hierarchical level and should not conflict with other requirements.

2. Derived requirements should be consistent with the system requirements or requirements from which they are derived.

3. The requirements should be consistent with the SSA.

4. The derived safety requirements should be defined and fed back to the SSA.

5. The requirements should be compatible with relevant hardware design standards.

6. The requirements should be compatible with the capabilities and limitations of available technology.

7. The component's requirements, such as performance, temperature range, de-rating and screening, should be consistent with the safety and reliability requirements.

8. The ability to test, maintain and manufacture the hardware item should be addressed.

9. The software/hardware interface requirements should be defined.

10. The requirements should be traceable upward to the next hierarchical level according to the criteria defined in the plan.

11. The derived requirements should capture the implementation constraints that will not be verified at a higher hierarchical level.

12. Omissions and errors should be fed back to the appropriate process for resolution.

Note 1: The following questions may help assess completeness of requirements:

a. Are all upper level requirements considered?

b. Are applicable standards and guidance considered?

c. Are all hardware functions and interfaces covered?

d. Is the architecture covered completely?

e. Is all of the hardware implementation requiring verification adequately specified?

f. Are all prohibited behavior characteristics in the safety assessment covered?

g. Is the operating environment adequately specified?

h. Are assumptions and constraints considered?

i. Will this implementation avoid any known problems with existing or similar hardware?

Note 2: The following questions may help assess correctness of requirements:

a. Are the requirements in accordance with upper level requirements?

b. Are the requirements in accordance with the system requirements allocated to the hardware item?

c. Are the requirements stating "what" as opposed to "how"?

d. Are the requirements unambiguous?

e. Can the requirements be realized?

f. Can the requirements be verified?

g. Have the functioning modes been defined?

h. Are the requirements consistent with the safety assessment?

i. Are assumptions and constraints correctly identified as derived requirements?

6.3.3.2 Design Review

A design review is a method to determine that the design data and implementation satisfy the requirements. Design reviews should be performed as defined in the plan at multiple times during the hardware design life cycle. Examples are conceptual design, detail design and implementation reviews. For hierarchical designs that span several hardware item levels, such as ASICs and circuit card assemblies, design reviews should be considered where the potential is greatest for assuring a correct design.

Guidance for design reviews includes:

1. All requirements should be addressed and the derived requirements and the design data should be correctly defined.
2. Environmental requirements should be addressed.
3. Safety and reliability requirements should be addressed.
4. The safety aspects of the design data should be explicitly identified.
5. The design should be capable of being implemented, tested and maintained.
6. New manufacturing techniques should be evaluated.
7. The components selection criteria identified in the plans should be satisfied.
8. The design should be traceable to the requirements.
9. Omissions and errors should be fed back to the appropriate process for resolution.

This Page Intentionally Left Blank

7.0 CONFIGURATION MANAGEMENT PROCESS

The configuration management process is intended to provide the ability to consistently replicate the configuration item, regenerate the information if necessary and modify the configuration item in a controlled fashion if modification is necessary. This section describes the objectives for hardware configuration management and activities that support those objectives.

7.1 Configuration Management Objectives

The objectives of the configuration management process are:

1. Configuration items are uniquely identified and documented.
2. Consistent and accurate replication of configuration items is ensured.
3. A controlled method of identifying and tracking modification to configuration items is provided.

7.2 Configuration Management Activities

Guidance for the configuration management activities includes:

1. Configuration items should be uniquely identified, documented and controlled. This may include, but is not limited to, hardware, design representations of hardware, tools or other data items used for certification credit and baselines.
2. Baselines should be established.
3. Problems should be uniquely identified, tracked and reported.
4. Change control and traceability of changes should be maintained. This requires that life cycle data identified in the plans should be secure and retrievable.
5. Archiving, retrieval and release of configuration items should be controlled.

Various methods may be used to satisfy configuration management objectives and activities and the following paragraphs provide guidance on activities that may be used as an acceptable method.

7.2.1 Configuration Identification

The purpose of the configuration identification activity is to label unambiguously each configuration item so that a basis is established for the control and reference of configuration items.

Guidance includes:

1. Configuration identification should be established for data items.
2. Configuration identification should be established for each configuration item, for each separately controlled component of a configuration item and for combinations of

configuration items that make up a product consistent with the plans agreed to by the certification authority.

Note: *The detail to which components, such as ASICs, configured PLDs, printed circuit boards and black boxes, are identified is determined by the Configuration Management Plan.*

3. Configuration identification should be established for COTS components and previously developed hardware items before they are used in a baseline.

4. Configuration identification should be established for each configuration item before it is used in a new baseline, referenced by other data items or used for product manufacturing.

7.2.2 Baseline Establishment

The purpose of baseline establishment is to define a basis for further activities and allow reference to, control of and traceability between configuration items.

Guidance includes:

1. Baselines should be established for configuration items used for certification credit.

   Note: *Intermediate baselines may be established to aid in controlling hardware activities.*

2. Once a baseline is established it should be subject to change control procedures.

3. Change control guidance should be followed when developing a derivative baseline from an established baseline.

4. If in developing a new baseline, certification credit is sought for activities or data associated with design of a previous baseline, this new baseline should be traceable to the previous baseline from which it was derived.

   Note: *The baseline may be a configuration item, a previously certified hardware item or a COTS component.*

7.2.3 Problem Reporting, Tracking and Corrective Action

The purpose of problem reporting, tracking and corrective action is to record problems and ensure correct disposition and resolution. Problems may include non-compliance with plans and standards, deficiencies of life cycle process outputs, anomalous behavior of products, and inadequacy or deficiency of tools and technology processes. Problem reporting should be implemented no later than the establishment of the baseline from which certification credit is to be obtained.

Guidance includes:

1. Each reported problem should be covered by a problem report.

2. Problem reporting should identify the configuration of the affected configuration items.

3. Problem reports that require corrective action should invoke the change control activity.

4. All closed problem reports should include a description of action taken to close the problem report including the completion of data item changes that were needed to implement a corrective action.

5. Not all problem reports have to be closed in order to obtain certification, however, all problem reports should be evaluated and those that are determined to have safety or certification impact should be closed.

6. The problem reporting system should track the status of problem reports, including their approval and disposition.

7.2.4 Change Control

The purpose of the change control activity is to ensure the recording, evaluation, resolution and approval of changes. Change control should be implemented in compliance with the configuration management plan and should be started no later than the establishment of the baseline from which certification credit is to be obtained.

Guidance includes:

1. Change control should preserve the integrity of the configuration items by providing protection against unauthorized change.

2. Change control should ensure that a change is assessed to determine whether or not the configuration identity needs to be updated.

3. Changes to configuration items under change control should be recorded, approved, and tracked. Approval authority is defined in the configuration management plan.

   Note 1: Problem reporting is related to change control, since resolution of a reported problem may result in changes to configuration items.

   Note 2: It is generally recognized that early implementation of change control assists the control and management of process activities.

4. Change control should ensure traceability of changes to the reason for the change.

5. Change control should ensure that the impact of the change is assessed to determine the effect of the change on the outputs of the processes and that the output data is updated.

   Note 1: Some or all of the activities of the processes may need to be repeated from the point at which their outputs are affected.

Note 2: *It should be recognized that a change to the manufacturing tools, technology processes or external components may impact the design.*

6. Change control should ensure that feedback is provided to affected processes.

7.2.5 Release, Archive and Retrieve

The purpose of the release activity is to place data items under configuration management control to ensure that only authorized data is used in other activities. The purpose of the archive and retrieve activity is to ensure that data items associated with the product can be retrieved in case of a need to duplicate, regenerate, re-test or modify the product.

Guidance includes:

1. Configuration items should be identified and released prior to use for manufacture and the authority for their release should be established.

2. Data items associated with the product should be retrievable from an approved source, such as the developing organization or company.

   Note: *Change control data and problem report data are part of the data items.*

3. Data retention procedures should be available to satisfy airworthiness requirements and enable modifications.

4. Procedures should be established to ensure the integrity of the stored data for as long as required by the certification authorities by:

   a. Ensuring that no unauthorized changes are made.

   b. Selecting storage media.

   c. Maintaining availability of stored data. For example, by exercising or refreshing archived data at a frequency compatible with the storage life of the medium.

   d. Ensuring that a single event that can cause irretrievable loss of archived data is unlikely. For example, by storing duplicate copies in physically separate archives.

7.3 Data Control Categories

Two categories associated with the configuration management of data items are defined: hardware control category 1 (HC1) and hardware control category 2 (HC2). Specifying two categories allows a less stringent configuration control for certain data items. HC1 requires all configuration management activities to be performed while HC2 is less restrictive. Data items classified as HC2 are not expected to change incrementally, but will be superceded by new data.

Table 7-1 defines the configuration management activities that are to be performed under HC1 and HC2. For example, Table 7-1 shows that data items identified in Appendix A, Table A-1 as HC2 need to be retrievable but do not need to be released. Additionally, Table 7-1 shows that any HC1 data item will have a baseline.

Appendix A identifies the control category for each data item as a function of hardware design assurance level. For example, in Table A-1, HC1 applies to hardware requirements for all assurance levels while HC2 applies to hardware review and analysis results for all assurance levels.

Table 7-1 Configuration Management Process Activities Associated with HC1 and HC2

Reference	Configuration Management Activity	HC1	HC2
7.2.1	Configuration Identification	x	x
7.2.2 (1),(2),(3)	Baselines	x	
7.2.2 (4) [1]	Baseline Traceability	x	x
7.2.3	Problem Reporting	x	
7.2.4 (1),(2)	Change Control - integrity and identification	x	x
7.2.4 (3),(4),(5),(6)	Change Control – records, approvals and traceability	x	
7.2.5 (1)	Release	x	
7.2.5 (2)	Retrieval	x	x
7.2.5 (3)	Data Retention	x	x
7.2.5 (4a)	Protection Against Unauthorized Changes	x	x
7.2.5 (4b),(4c),(4d)	Media Selection, Refreshing, Duplication	x	

[1] Identification of HC2 data for use with the new baseline does not imply reclassification of the data to HC1.

This Page Intentionally Left Blank

8.0 PROCESS ASSURANCE

Process assurance ensures that the life cycle process objectives are met and activities have been completed as outlined in plans or that deviations have been addressed. This section describes the objectives for process assurance and the activities that support those objectives. There is no intent to impose specific organizational structures.

Process assurance activities should be achieved with independence in order to objectively assess the life cycle process, identify deviations and ensure corrective action.

8.1 Process Assurance Objectives

The objectives of process assurance are to ensure that:

1. Life cycle processes comply with the approved plans.

2. Hardware design life cycle data produced complies with the approved plans.

3. The hardware item used for conformance assessment is built to comply with the associated life cycle data.

8.2 Process Assurance Activities

Guidance for the process assurance activities includes:

1. Availability of hardware plans as specified in the planning process section of this document and as agreed to in the PHAC should be ensured.

2. Holding of reviews in compliance with the approved plans and tracking of resulting action items to closure should be ensured.

3. Detection, recording, evaluation, approval, tracking and resolution of deviations from the hardware plans and standards should be ensured.

4. Satisfaction of the transition criteria of the hardware life cycle processes in compliance with the approved plans should be ensured.

   Note: *Audits are an effective method for performing activities in items 1 through 4 above.*

5. An inspection should be performed to ensure that the hardware item is built in compliance with its design data.

   Note: *An example of this activity is a First Article Inspection.*

6. Records of the process assurance activities, including evidence of assessment of completion of design activities, should be produced.

7. Where applicable, the applicant should ensure that the processes used by subcontractors are consistent with the hardware plans.

This Page Intentionally Left Blank

9.0 CERTIFICATION LIAISON PROCESS

The purpose of the certification liaison process is to establish communication and understanding between the applicant and the certification authority throughout the hardware design life cycle to assist in the certification process. The certification liaison process should be accomplished as described by the hardware planning process, Section 4, and the PHAC, Section 10.1.1. Table A-1 of Appendix A gives a summary of the outputs of this process. In addition, liaison activities may include design approach presentation for timely approval, negotiations concerning the means of compliance with the certification basis, approval of design approach, means of data approval, and any required certification authority reviews and witnessing of tests.

At completion of a project, a summary of the design processes followed, outputs produced and status of the hardware item should be described in the Hardware Accomplishment Summary, Section 10.9.

9.1 Means of Compliance and Planning

The applicant proposes a means of compliance for hardware. The PHAC defines the proposed means of compliance. Guidance includes:

1. The PHAC, hardware verification plan and other requested data should be submitted to the certification authority for review at a point in time when the effects of design changes on the program are minimal.

2. Issues identified by the certification authority concerning the planning for the hardware aspects of certification should be resolved.

3. Agreement on the PHAC should be obtained with the certification authority.

4. Liaison with the certification authority during the design and certification cycle as outlined in the plan should be continued and issues raised by the certification authority resolved in a timely manner.

In some programs, the certification liaison is not provided by the equipment manufacturer, but by the airframe or other customer with the equipment manufacturer in a supporting role. This relationship should be defined in the PHAC and contact with the certification authority should be through the applicant for certification. It is the responsibility of the applicant for certification to ensure that data is provided to the certification authority.

When some hardware items embedded in the equipment are procured from a subcontractor, the certification plan should identify which data are expected from the subcontractor and which are to be generated by the applicant.

It is acceptable for an applicant to include the PHAC and verification plan with other related plans within the top-level certification plan

9.2 Compliance Substantiation

The applicant provides evidence that the hardware design life cycle processes have satisfied the hardware plans. Certification authority reviews may take place at the applicant's facilities or applicant's supplier's facilities. The applicant arranges these reviews and makes hardware design life cycle data available as needed.

The applicant should:

1. Resolve issues raised by the certification authority as a result of its reviews.

2. Submit the Hardware Accomplishment Summary, Section 10.9 and Top Level Drawing, Section 10.3.2.2.1 to the certification authority.

3. Submit or make available other data or evidence of compliance requested by the certification authority.

10.0 HARDWARE DESIGN LIFE CYCLE DATA

This section describes the hardware design life cycle data items that may be produced during the hardware design life cycle for providing evidence of design assurance and compliance with certification requirements. The scope, amount and detail of the life cycle data needed by the certification authorities as design assurance evidence will vary depending on a number of factors. These factors include the applicable certification authority requirements for the airborne system, the assigned design assurance levels, the complexity and the service experience of the hardware. Details of the design assurance evidence should be identified, recorded in the PHAC and agreed to with the certification authorities.

The additional considerations in Section 11 and the design assurance considerations for Level A and B functions in Appendix B may lead to the generation of additional life cycle data.

Appendix A indicates the hardware design life cycle data to be developed, the degree of verification independence, and the applicable data control category, as defined in Section 7, in terms of the hardware design assurance level.

1. The hardware design life cycle data characteristics should be:

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

   b. **Complete.** Information/data includes necessary and relevant requirements and descriptive material, labeled figures, and defined terms and units of measure.

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

   d. **Consistent.** Information/data contains no conflicts.

   e. **Modifiable.** Information/data is structured and changes can be made completely, consistently and correctly while retaining the structure.

   f. **Traceable.** Information/data origin can be determined.

   The descriptions of this section are not intended to imply a particular data packaging method, form or organization of the hardware life cycle data within a package. For example, all plans, standards, and procedures may be described in a single document or multiple documents.

2. The data packaging method, form and organization should be proposed in the PHAC and agreement with the certification authority obtained early in the program.

3. Agreed-upon information and data should be retrievable and available throughout the service life of the airborne system or equipment.

10.1 Hardware Plans

The hardware plans describe the processes, procedures, methods, and standards to be used for the hardware certification, design, validation, verification, process assurance and configuration control.

10.1.1 Plan for Hardware Aspects of Certification

The PHAC defines the processes, procedures, methods and standards to be used to achieve the objectives of this document and obtain certification authority approval for certification of the system containing hardware items. The PHAC, once approved, represents an agreement between the certification applicant and the certification authority on the processes and activities to be conducted and the resultant evidence to be produced to satisfy the hardware aspects of certification. The PHAC may be part of another plan, such as the airborne system certification plan.

The PHAC should include:

1. **System Overview.** This section provides an overview of the airborne system in which the hardware items are to be used, including a system functional description, system failure conditions, system architecture, a description of the allocation of the functions to hardware items and software, and references to existing system documentation.

2. **Hardware Overview.** This section describes the hardware functions, hardware items, architecture, new technologies to be used, and any fail-safe, fault tolerant, redundancy and partitioning techniques to be used.

3. **Certification Considerations.** This section describes the certification basis, proposed means of compliance and the hardware design assurance level of each function of the hardware item. It also provides the justification for the hardware design assurance level assignment based on a safety assessment of the hardware and its use within the airborne system, including a description of potential hardware failure conditions as discussed in Section 2.3.4. When applicable, either a summary of the FFPA or plan for performing an FFPA and applying the results should also be included.

4. **Hardware Design Life Cycle.** This section describes the procedures, methods and standards to be applied and processes and activities to be performed to meet the hardware design assurance objectives. It describes the activities, combinations and sequencing of activities, relationships between processes and activities, transition criteria, responsibilities, tool usage, and means for providing feedback and interaction among hardware processes and between hardware processes and the system and software processes. This section may reference applicable plans, policies, standards, procedures and deviations to those plans and standards for the program.

5. **Hardware Design Life Cycle Data.** This section describes or references the data to be developed and submitted or available as evidence of compliance to the objectives of this document and the plan.

6. **Additional Considerations.** This section describes the additional considerations. These include use of previously developed hardware, including references to applicable data to be reused, COTS usage, product service experience, and tool assessment and qualification as described in Section 11, or design assurance considerations for Level A or B functions as described in Appendix B.

7. **Alternative Methods**. This section describes any alternative methods proposed for the program which are either not described in this document or are to be applied in a manner other than as described in this document. Justification for why the alternative method is acceptable should be provided.

8. **Certification Schedule.** This section identifies the major program milestones and the dates when hardware design life cycle data will be submitted to the certification authority.

10.1.2 Hardware Design Plan

The hardware design plan describes the procedures, methods and standards to be applied and the processes and activities to be conducted for the design of the hardware item. This plan may be included in the PHAC and may reference design policies and standards to be applied.

The hardware design plan should include:

1. **Hardware Design Life Cycle.** References to design policies and standards to be applied and a description of the hardware design life cycle processes and activities that will be used to achieve the design objectives for the hardware design assurance level.

2. **Hardware Product Description.** Identification of the hardware specifications to be achieved, alternative uses, planned service life and upgrade considerations.

3. **Hardware Design Methods.** Description of the requirements capture and specification methods, conceptual design methods, detailed design methods, synthesis techniques, implementation methods, and production transition methods to be used for the hardware item. When architectural mitigation for Level A or B functions, as described in Appendix B, Section 3.1, has been considered but not finalized at the time this plan is written, state how the decision will be carried into the design process.

4. **Hardware Design Environment.** Description of the design tools to be used.

5. **Hardware Item Data.** Identification of hardware item design data to be produced or references to previously developed hardware item specifications, document and drawing numbers, and part numbers.

6. **Other Considerations.** Description of planned process technology options, use and assembly options, product packaging, and hardware mounting options.

10.1.3 Hardware Validation Plan

The validation 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 may be included in the PHAC and may reference validation standards to be applied.

The validation plan should include:

1. **Validation Methods.** Description of and references to the validation procedures, standards and methods to be used. Methods may include analyses, reviews and testing.

2. **Validation Data.** Identification and description of the evidence to be produced as a result of the hardware validation process.

3. **Validation Environment.** Identification and description of analysis and test equipment and validation tools to be used to implement the validation process and activities.

10.1.4 Hardware Verification Plan

The verification plan 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. This plan may be included in the PHAC and may reference verification policies and standards to be applied.

The verification plan should include:

1. **Verification Methods.** Description of and references to the verification policies, procedures, standards and methods to be used to provide objective evidence of the integrity of the hardware items, including COTS and unused functions. Methods may include analyses, reviews and testing. When the advanced analysis methods of Appendix B, Section 3.3 are employed, include a detailed description of the methods for the applicable FFPs and the applicable verification completion criteria.

2. **Verification Data.** Identification and description of the evidence to be produced as a result of the hardware verification process.

3. **Verification Independence.** Description of the means to be used to assure verification independence for those objectives requiring independence.

4. **Verification Environment.** Identification and description of analysis and test equipment and verification tools to be used to implement the verification process and activities.

5. **Organizational Responsibilities.** Identification of the organizations responsible for implementing the verification process.

10.1.5 Hardware Configuration Management Plan

The hardware configuration management plan describes the policies, procedures, standards and methods to be used to satisfy the configuration management objectives of this document.

The hardware configuration management plan should include:

1. **Hardware Configuration Management Methods.** Description of and reference to the policies, procedures, standards and methods to be used to identify, manage, and control the hardware and its life cycle data.

2. **Hardware Baselines.** Description of the methods and procedures used to establish design and product baselines and provide baseline traceability.

3. **Problem Reporting and Resolution.** Description of the methods and procedures to be used for recording, tracking and resolving problem reports.

4. **Change Control.** Description of the methods, procedures and processes for identifying, controlling, and tracking changes to controlled data items.

5. **Storage and Retrieval.** Description of the procedures for release, archival and retrieval of hardware design life cycle data. The description should include archive content, format, and medium standards, rules, methods and criteria.

6. **Environment Control.** Description of the procedures and method for identifying and controlling the tools used for developing and verifying the hardware.

7. **Configuration Management Tools.** Description of the tools and resources used for the configuration management process and activities.

10.1.6 Hardware Process Assurance Plan

The hardware process assurance plan describes the procedures, methods and standards to be applied and the processes and activities to be conducted for achieving the process assurance objectives of this document.

The hardware process assurance plan should include:

1. **Process Control.** Description of the policies and procedures for implementation of process assurance of the hardware design processes.

2. **Organizational Responsibilities.** Identification of the organizations responsible for implementing process assurance.

3. **Conformance.** Description of the policies, procedures and criteria for determining process and product conformance.

4. **Process Assurance Activities.** Description of the process assurance reviews and audits to be conducted to demonstrate compliance of the processes to plans and standards.

5. **Deviations.** Description of the methods for detecting, recording, evaluating, resolving and approving deviations from plans and standards.

10.2 Hardware Design Standards and Guidance

Hardware design standards and guidance may define the rules, procedures, methods, and criteria for hardware design, validation, verification, assurance and control processes and are used to assess the acceptability and quality of hardware design results. Standards may not be required, but, if the applicant invokes them for the project, they become part of the certification basis and plans for the project. As with the plans, such standards and guidance may be packaged as a single document or multiple documents. Tools may be used to enforce standards.

10.2.1 Requirements Standards

Requirements standards may be used during the requirements capture process to define the rules, procedures, methods, guidance and criteria for developing the requirements. Requirements standards may include methods and criteria for developing and specifying requirements, methods and criteria for validating the requirements, notations used to express the requirements, guidance on the use of requirements specification tools, and the means used to provide derived requirements to the system design process.

10.2.2 Hardware Design Standards

Hardware design standards may be used during the conceptual design process and detailed design process to define the rules, procedures, methods, guidance and criteria for developing and specifying the hardware design.

Hardware design standards may include:

1. Hardware design representation methods and notations.

2. Design specification and naming conventions.

3. Guidance on design methods.

4. Guidance on the use of hardware design tools.

5. Guidance for electronic component selection.

6. Guidance for assessing design alternatives.

7. Guidance for assessing the fail-safe and fault-tolerance design constructs.

8. Description of the means for providing feedback to the requirements process and for clarifying requirements.

10.2.3 Validation and Verification Standards

Hardware validation and verification standards may be used during the validation and verification processes to define the rules, procedures, methods, guidance and criteria for validating and verifying the hardware design and implementation.

10.2.4 Hardware Archive Standards

Hardware archive standards may be used to define the procedures, methods and criteria used to retain and archive product data and develop and maintain program and project archives. Hardware archive standards may include archive content, format, and medium standards, rules, methods and criteria.

10.3 Hardware Design Data

The hardware design data are the specifications, documents and drawings that define the hardware items.

10.3.1 Hardware Requirements

The requirements specify the functional, performance, safety, quality, maintainability, and reliability requirements for the hardware item being developed.

The requirements should include:

1. The system design and safety requirements allocated to the hardware.

2. Identification of applicable standards for the hardware.

3. Hardware functional and performance requirements, including derived requirements and stress limits for normal use.

4. Hardware reliability and quality requirements, including requirements related to failure rates, exposure times and design constraints.

5. Hardware maintenance and repair requirements throughout the hardware item service life.

6. Hardware manufacturability and assembly requirements.

7. Hardware testability requirements.

8. Hardware storage and handling requirements.

9. Installation requirements.

10.3.2 Hardware Design Representation Data

The hardware design representation data provides a definition of the hardware item and is comprised of the set of drawings, documents and specifications used to build the hardware item. The following paragraphs define some typical hardware design data and their content. The type of data, drawings and documents produced for a given hardware design will vary depending on the size, complexity and number of components the hardware item contains.

10.3.2.1 Conceptual Design Data

The conceptual design data is the data that describes the hardware item's architecture and functional design and may include:

1. A high-level description, such as a block diagram or HDL definition, which outlines the major functions and shows the flow of information between these functions.

2. The mechanical structure which describes the arrangement of the hardware item, such as drawings or sketches showing exterior package, printed circuit board arrangement, connector selection and location, and major interconnect wiring.

3. Other architectural features and partitioning that are important from an airworthiness point of view. This might include items such as EMI, lightning, shock or vibration protection, unused functions in major components as well as man-machine interfaces, such as ergonomic factors, lighting characteristics and display resolution.

4. Top-level hardware item functional description.

5. Hardware item functional architecture.

6. Preliminary hardware safety assessment data.

10.3.2.2 Detailed Design Data

The detailed design data describes the data necessary to implement the hardware item consistently with its requirements. Depending on the hierarchical level of the hardware item, this may include top-level drawing, assembly drawings, interconnection data, parts data, HDL hardware description, reliability data, test methodology data, list of unused functions in selected components and actions taken to assure they will not compromise the safety of the hardware item, installation control data, and hardware/software interface data. Some specific data are described below.

Note: *In addition to the detailed design data required by other applicable certification requirements, such as Technical Standard Orders, the content and availability of other detailed design data items are proposed by the applicant to the certification authority in the PHAC.*

10.3.2.2.1 Top-Level Drawing

The top-level drawing uniquely identifies the hardware item and identifies all assemblies, subassemblies, components and relevant documentation that define the hardware item.

10.3.2.2.2 Assembly Drawings

Assembly drawings include additional detailed information needed to assemble the hardware item, assembly, or subassembly.

An assembly drawing may include:

1. Location and orientation of the hardware items within a hardware assembly.
2. Identification of assembly instruction sequences or methods to ensure a correct and fault free assembly.
3. Locations for identifying marks, labels, vision references used in subsequent operations.

10.3.2.2.3 Installation Control Drawings

Installation control drawings ensure correct installation of a hardware item into a system or correct installation of a hardware item into another hardware item. For some lower level hardware item, assembly drawings for the next higher hardware item or assembly may act as the installation control drawing.

Installation control drawing may include:

1. Dimensions.
2. Clearance requirements.
3. Cooling and mounting information.
4. Information on weight, center of gravity, and other parameters necessary to ensure safe and proper installation.

10.3.2.2.4 Hardware/Software Interface Data

The performance of the hardware as determined by the requirements specification may depend upon the configuration of the hardware by the software, calibration of the hardware by the software or upon a necessary interaction between the hardware and software.

Data relating to the interface between the hardware and the software may include:

1. Memory addresses.
2. Allocation of memory address fields into which data can be loaded.
3. Timing and sequence information.
4. Other information necessary for the operation of the hardware/software interface.

10.4 Validation and Verification Data

Validation and verification data is the evidence of the completeness and correctness of the hardware design results and of the hardware item itself. It provides assurance that the hardware has been developed to its requirements and design, correctly produced, and the design objectives achieved. Data includes procedures and results for hardware reviews, analyses and testing. Additional data items beyond that described in this section may be needed for Level A and B functions as described in Appendix B.

10.4.1 Traceability Data

Hardware traceability establishes a correlation between the requirements, detailed design, implementation and verification data that facilitates configuration control, modification and verification of the hardware item.

Hardware traceability data should include:

1. A correlation between the system requirements allocated to hardware and the requirements.
2. A correlation between the requirements and the hardware detailed design data.
3. A correlation between the hardware detailed design data and the as-built hardware item or assembly.
4. A correlation between the requirements, including derived hardware requirements, and detailed design data and the verification procedures and results.
5. The results of a traceability analysis.

10.4.2 Review and Analysis Procedures

Hardware review and analysis procedures define the process and criteria for conducting reviews and analyses.

Hardware review and analysis procedures should include:

1. Purpose of review or analysis.
2. Organizations to participate in the review.
3. Review or analysis criteria.
4. Detailed instructions for conducting the review or analysis.
5. Review or analysis acceptability and completion criteria.

10.4.3 Review and Analysis Results

Hardware review and analysis results are the evidence that the reviews and analyses have been completed to approved procedures and criteria.

Hardware review and analysis results should include:

1. Identification of review or analysis procedure.
2. Identification of data item reviewed or analyzed.
3. Personnel participating in the review or analysis.

4. Review or analysis results.

5. Corrective actions generated as a result of review or analysis, such as listing of problem reports or action items.

6. Review or analysis conclusion including, for reviews, a qualitative assessment of the item reviewed and, for analysis, a quantitative assessment of the item analyzed and the analysis data.

10.4.4 Test Procedures

Hardware test procedures define the methods, environment and instructions for conducting both functional and environmental qualification testing used for the verification of the hardware item.

Hardware test procedures should include:

1. Purpose of test.

2. Identification of the hardware test setups, software and test equipment setup instructions required for each hardware test.

3. Detailed instructions for conducting the test procedures.

4. Test input data.

5. Expected results, such as pass/fail criteria and requirements covered by the test.

10.4.5 Test Results

Hardware test results are the objective evidence that the tests have been completed to approved procedures in support of the verification of the hardware item.

Hardware test results should include:

1. Identification of the test procedure.

2. Identification of the item tested.

3. Actual results of conducting the test.

4. Identification of the personnel conducting and witnessing the tests, if applicable, and the date the tests were conducted.

5. Interpretation of results, either by analysis or review and actual test coverage achieved.

10.5 Hardware Acceptance Test Criteria

This data provides the criteria and assessment data that the test and associated test results are capable of ensuring that an item is manufactured or repaired correctly.

The criteria should include:

1. Key attributes to be tested.
2. Pass/fail criteria for each key attribute.
3. Any test constraints.
4. Substantiation of the key attributes and pass/fail criteria.
5. Coverage of design aspects necessary to meet the safety requirements.
6. Assessment data that shows that the test criteria have been properly implemented based on the actual test procedures and associated test results.

10.6 Problem Reports

Problem reports are a means to identify and record the resolution to hardware design problems, process non-compliance with hardware plans and standards, and deficiencies in hardware life cycle data.

Problem reports should include:

1. Identification of the configuration item and process activity in which the problem was observed.
2. Identification of the configuration items to be modified or a description of the process to be changed.
3. A problem description which enables the problem to be understood and resolved.
4. A description of the corrective action taken to resolve the reported problem.

10.7 Hardware Configuration Management Records

The results of the configuration management process activities are recorded in configuration management records. These may include configuration identification lists, baseline or electronic records, change history reports, problem report summaries, tool identification data, archive records and release records.

10.8 Hardware Process Assurance Records

The results of the process assurance process activities are recorded in process assurance records. These may include review or audit reports, meeting minutes, records of authorized process deviations, or conformity review records.

10.9 Hardware Accomplishment Summary

The Hardware Accomplishment Summary is the primary data item for showing compliance to the PHAC and demonstrating to the certification authority that the objectives of this document have been achieved for the hardware items. This summary may be combined with the system accomplishment summary. The Hardware

Accomplishment Summary should include the following information as documented in the PHAC:

1. System overview.
2. Hardware overview.
3. Certification considerations.
4. Hardware design life cycle description.
5. Hardware design life cycle data.
6. Previously developed hardware.
7. Additional considerations.
8. Alternative methods

Differences from the approved PHAC should be identified. In addition, the following four items should be addressed:

1. **Hardware Identification.** This section identifies the hardware configuration and hardware items by part number and version.
2. **Change History.** If applicable, this section includes a summary of hardware changes with attention to changes made due to failures affecting safety, and identifies changes from the hardware design life cycle processes since the previous certification.
3. **Hardware Status.** The section contains a summary of problem reports unresolved at the time of certification, including a statement of functional limitations.
4. **Compliance Statement.** This section includes a statement of compliance with this document and a summary of the methods used to demonstrate compliance with criteria specified in the hardware plans. This section also addresses additional rulings and deviations from the hardware plans, procedures, and this document.

   Note: *The data included in the PHAC does not necessarily need to be repeated in the Hardware Accomplishment Summary, however doing so may expedite the certification process.*

This Page Intentionally Left Blank

11.0 ADDITIONAL CONSIDERATIONS

This section provides guidance on additional considerations of design assurance that are not covered in the previous sections. These additional considerations may be used at the applicant's discretion to satisfy some of the objectives of Section 2 through Section 9. Any use of additional considerations should be agreed with the certification authority.

11.1 Use of Previously Developed Hardware

This section discusses the issues associated with the use of previously developed hardware. Guidance includes the assessment of modifications to the hardware, to the aircraft installation, to the application environment, or to the design environment and upgrading design baselines. Guidance for COTS component usage, a special case of previously developed hardware, is covered in Section 11.2. Configuration Management and Process Assurance considerations should also be addressed for each use of previously developed hardware.

The intention to use previously developed hardware should be stated in the PHAC.

11.1.1 Modifications to Previously Developed Hardware

This section discusses modifications to previously developed hardware. Modification may result from requirement changes, the detection of errors, hardware or technology enhancements, or procurement difficulties.

Analysis activities for proposed modifications include:

1. Review of the outputs of the system safety assessment process.

2. Application of the guidance of Section 11.1.4 if the hardware design assurance level is increased.

3. The impact of changes should be analyzed, including the consequences of changes that may result in a re-verification effort involving more than the area changed. This area may be determined by signal flow analysis, functional analysis, timing analysis, traceability analysis or other suitable means.

11.1.2 Change of Aircraft Installation

This section discusses the use in a new aircraft installation of hardware that has been previously certified at a certain hardware design assurance level and under a specific certification basis. When using previously developed hardware on new aircraft installations, the following guidance should be used:

1. The system safety assessment process assesses the new aircraft installation and determines the hardware design assurance level and the certification basis. No additional effort will be required if these are the same or less stringent for the new installation as they were in the previous installation.

2. If functional modifications are required for the new installation, the guidance of Section 11.1.1, Modifications to Previously Developed Hardware, should be satisfied.

3. If the previous design activity did not produce the outputs required to substantiate the safety objectives of the new installation, the guidance of Section 11.1.4, Upgrading A Design Baseline, should be satisfied.

11.1.3 Change of Application or Design Environment

Use of previously developed hardware may involve a new design environment, or integration with other software or hardware than that used for the original application.

New design environments may increase or reduce some activities within the hardware design life cycle processes. Guidance includes:

1. If a new design environment uses hardware design tools, the guidance of Section 11.4, Tool Assessment and Qualification, may be applicable.

2. Verification of hardware interfaces should be conducted where previously developed hardware is used with different interfacing hardware.

3. The need for re-verification of hardware/software interfaces should be addressed when previously developed hardware uses different software.

11.1.4 Upgrading a Design Baseline

The following guidance is for hardware items whose life cycle data from a previous application are determined to be deficient for the safety objectives associated with a new application. This guidance is intended to aid the applicant in obtaining agreement with the certification authority for hardware previously developed at a lower hardware design assurance level:

Guidance for upgrading a design baseline includes:

1. The objectives of this document should be satisfied, while taking advantage of life cycle data of the previous development.

2. Hardware aspects of certification should be based on the failure conditions and hardware design assurance levels as determined by the system safety assessment process. The impact of the changes to the previous application should be analyzed to determine areas of deficiency.

3. Life cycle data from a previous development should be evaluated to ensure that the verification process objectives are satisfied for the hardware that is planned for implementation of the upgraded function at the required hardware design assurance level.

4. Reverse engineering may be used to regenerate hardware life cycle data that is deficient or missing to satisfy the design assurance objectives of this document

5. If use of product service experience is planned to satisfy the design assurance objectives of this document in upgrading a design baseline, the guidance of Section 11.3, Product Service Experience, should be addressed.

6. The applicant should specify the strategy for accomplishing compliance with this document in the PHAC.

11.1.5 Additional Configuration Management Considerations

The configuration management process for the new application of previously developed hardware should include, in addition to the guidance of Section 7:

1. Traceability from the hardware product and life cycle data of the previous application to the new application.

2. Change control processes that can manage change requests from different applications of the common item.

11.2 Commercial-Off-The-Shelf (COTS) Components Usage

COTS components are used extensively in hardware designs and typically the COTS components design data is not available for review. The certification process does not specifically address individual components, modules, or subassemblies, as these are covered as part of the specific aircraft function being certified. As such, the use of COTS components will be verified through the overall design process, including the supporting processes, as defined in this document. The use of an electronic component management process, in conjunction with the design process, provides the basis for COTS components usage.

11.2.1 Electronic Component Management for COTS Components

Electronic component management for COTS components is an important supporting process associated with the design and development of hardware. The processes of electronic component management apply to COTS electronic components. While there are both business and technical aspects of this process, this section only deals with the technical aspects as they impact certification.

Certification credit may be gained by establishing that:

1. The component manufacturer can demonstrate a track record for production of high quality components.

2. Quality control procedures are established at the component manufacturer.

3. There is service experience supporting the successful operation of the component.

4. The component has been qualified by the manufacturer or by means of additional testing, which establish the component reliability.

5. The component manufacturer has control of the component quality level or that this is assured by means of additional component testing.

6. The components have been selected on the basis of technical suitability of the intended application, such as component temperature range, power or voltage rating, or that additional testing or other means has been used to establish these.

7. The component performance and reliability are monitored on a continuous basis, with feedback to component manufacturers concerning areas that need improvement.

11.2.2 COTS Component Procurement

COTS component procurement guidance is not the intent of this document but feedback of procurement issues should be managed and resolved by the applicant when they have significant impacts on hardware design assurance.

Major concerns include:

1. Actual availability of COTS component design assurance data as required by this document.

2. Variations in component parameters that depend on production batches may not be identified, even by robustness tests.

3. Evolving aspects of electronic component technology.

4. COTS components which become non-procurable.

11.3 Product Service Experience

Service experience may be used to substantiate design assurance for previously developed hardware and for COTS components. Service experience relates to data collected from any previous or current usage of the component. Data from non-airborne applications is not excluded.

Note: *Wide and successful use of an item in service may provide confidence that the item's design is mature and free of errors and that the manufacturing quality of the item is demonstrated.*

11.3.1 Product Service Experience Data Acceptability Criteria

When service experience data is used for design assurance, the relevance and acceptability of the service experience data depends on one or more of the following:

1. Similarity of hardware item usage with respect to application, function, operating environment and design assurance level.

2. Extent to which the design assurance data is based on the proposed configuration of the hardware item.

3. Extent to which the design errors found during the service period being assessed have been eliminated, mitigated, or analyzed and determined to have no safety impact in the configuration to be used.

4. Actual failure rates in operation.

   Note: *The PHAC should specifically address those aspects where the design assurance of parts of an application relies on service experience data.*

11.3.2 Assessment of Product Service Experience Data

To satisfy the above criteria the applicant should:

1. Assess the relevance of previous applications, installations and environments to the target application, based upon engineering analysis.

   Note: *Data used to determine appropriateness of use and usage limitations may be available in specifications, data sheets, application notes, service bulletins, user correspondence and errata notices. These sources of information may also describe the functions associated with the hardware item, so the airborne intended use can be correlated to previous uses.*

2. Assess the intended usage for impacts on the safety assessment process, including possible mitigation of the effects of design errors identified by the data.

3. Assess any available statistics on design errors and their impact on the safety assessment process. A qualitative assessment can be used if statistics are not available.

4. Assess available problem reports. Problem reports may show that service experience has led to improvements now available in the current configuration. Problems identified but not fixed may still be mitigated by architectural means or by performing additional verification. Establish or assess the relationships between problem reports and hardware item or product requirement changes.

   Note: *For electronic components, substantial service usage may increase the likelihood that errors have been detected and eliminated or that temporary "fixes" are available.*

11.3.3 Product Service Experience Assessment Data

Service experience assessment data used to substantiate the design assurance for the proposed application should include:

1. Identification of the component and its intended function in the airborne system. Identify the design assurance level, or for components used in Level A and B functions, a description of additional means of assurance for the component, such as architectural means and additional or advanced verification strategies to be applied.

2. A description of the service experience data collection and assessment process, including criteria for determining the adequacy and validity of the data.

3. The service experience data, including the detailed service information being considered, change history, assumptions used to analyze the service experience data and a summary of the analysis results.

4. Justification for the adequacy of the service experience data relative to the intended use and required design assurance level.

11.4 Tool Assessment and Qualification

Tools, both hardware and software, will normally be used during hardware design and verification. When design tools are used to generate the hardware item or the hardware design, an error in the tool could introduce an error in the hardware item. When verification tools are used to verify the hardware item, an error in the tool may cause the tool to fail to detect an error in the hardware item or hardware design. Prior to the use of a tool, a tool assessment should be performed. The results of this assessment and, if necessary, tool qualification should be recorded and maintained.

The purpose of tool assessment and qualification is to ensure that the tool is capable of performing the particular design or verification activity to an acceptable level of confidence for which the tool will be used.

11.4.1 Tool Assessment and Qualification Process

Tool assessment assesses the role of the tool in a design life cycle process and may include qualification activities to be performed depending on the role of the tool and design assurance level of the hardware function. This assessment guidance is presented as a flowchart and applies to both design tools and verification tools when used to meet objectives or generate data items to satisfy those objectives. The flowchart will lead the applicant to limited appraisal of some categories of tools and to tool qualification of others.

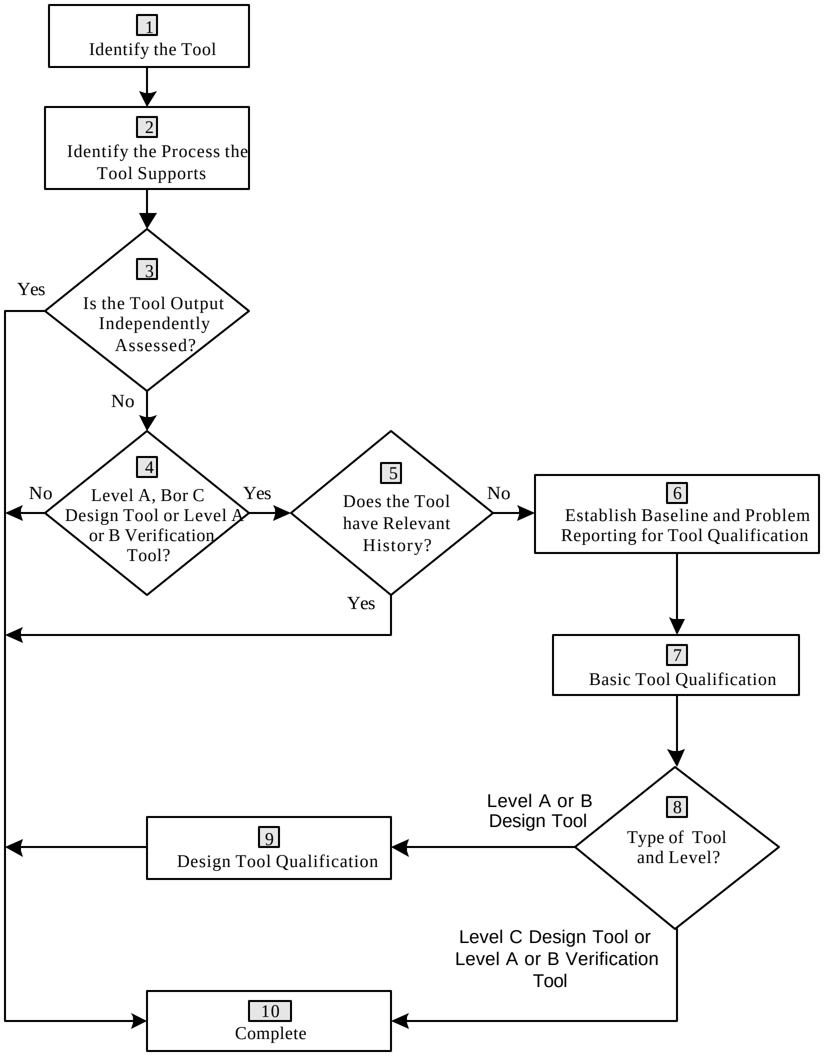
The tool assessment and qualification process may be applied to either a single tool or a collection of tools. Tools often contain capabilities beyond those needed for a specific design or verification activity on any specific project. It is only necessary to assess those functions of the tool used for a specific hardware life cycle activity, not the entire tool.

It is recognized that tools are often integrated and shared during the various life-cycle phases. If the same tool is used during both the design and the verification phase, then the tool may need to be assessed as a design tool unless separation of and protection between the two functions can be established.

<u>Note 1:</u> *If the assessment of a given tool indicates that some of its functions are used for design but other functions are used for verification, it may be worthwhile to address the functions separately and perform the assessment for each group of the tool's assessed functions.*

Note 2: This assessment activity focuses as much or more on the application of the tool as the tool itself.

The flow chart of Figure 11-1 indicates the tool assessment considerations and activities and provides guidance for when tool qualification may be necessary. The numbers in the decision and activity blocks refer to the numbered items following the figure that provide further clarification of the decision or activity.



1
Identify the Tool
2
Identify the Process the Tool Supports
3
Is the Tool Output Independently Assessed?
Yes
No
4
Level A, Bor C Design Tool or Level A or B Verification Tool?
No
Yes
5
Does the Tool have Relevant History?
No
Yes
6
Establish Baseline and Problem Reporting for Tool Qualification
7
Basic Tool Qualification
8
Type of Tool and Level?
Level A or B Design Tool
9
Design Tool Qualification
Level C Design Tool or Level A or B Verification Tool
10
Complete

Figure 11-1 Design and Verification Tool Assessment and Qualification

1. **Identify the Tool**. Includes the name, source, version number and the host environment on which it is based. Tool updates should be documented and tracked.

   Note: When updating a tool, assess the potential impacts of tool updates on existing results and on the remaining life cycle of the hardware.

2. **Identify the Process the Tool Supports.** Identify the design or verification process that the tool supports, any relevant limitations of the tool and the outputs it produces for use in the hardware design life cycle. If certain problems are known to exist with the tool, provide a statement of acceptability for use of the tool with justification.

3. **Is the Tool Output Independently Assessed?** An independent assessment verifies the correctness of the tool output using an independent means. If the tool output is independently assessed, then no further assessment is necessary.

   Note: *Independent assessment of a design tool's output that is generated in whole or in part by the tool may be established by the verification activities performed on the item, such as component, netlist or assembly. In this case, the integrity of the end item does not depend upon the correctness of the design tool output alone.*

   Independent assessment of a verification tool's output may include a manual review of the tool outputs or may include a comparison against the outputs of a separate tool capable of performing the same verification activity as the tool being assessed.

   The applicant may propose other methods of independent assessment as well.

4. **Is the Tool a Level A, B or C Design Tool or a Level A or B Verification Tool?** If the tool is used for Level D functions, as a verification tool for Level C functions, or used to assess the completion of verification testing, such as in an elemental analysis as described in Appendix B, Section 3.3.1.1.2, no further assessment is necessary. If the tool is used as a design tool for hardware implementing a Level A, B or C function or is used as a verification tool for hardware implementing a Level A or B function, then further assessment is needed.

5. **Does the Tool have Relevant History?** When it is possible to show that the tool has been previously used and has been found to produce acceptable results, then no further assessment is necessary. A discussion of the relevance of the previous tool usage versus the proposed usage of the tool should be included in the justification.

   Note: *The history of the tool may be based on either an airborne or non-airborne application, provided that data is available to substantiate the relevance and credibility of the tool's history.*

6. **Establish Baseline and Problem Reporting for Tool Qualification.** Establish a baseline for tool configuration management and tool problem reporting to prepare for tool qualification.

7. **Basic Tool Qualification.** Establish and execute a plan to confirm that the tool produces correct outputs for its intended application using analysis or testing. The

tool's user guide or other description of the tool's function and its use may be used to generate requirements.

8. **Type of Tool and Level?** Is the tool being considered a Level A or B hardware design tool or a Level C hardware design tool or a Level A or Level B hardware verification tool?

9. **Design Tool Qualification.** Qualify the Level A or B design tool using the strategies described in Appendix B of this document, the tool qualification guidance of RTCA DO-178B / EUROCAE ED-12B for software development tools or other means acceptable to the certification authority. Independence of this activity from the tool development should also be established.

   Note: *Specific guidance for Level A and B design tool qualification is not provided here because of the variability of the circumstances of the tool usage, technology involved, visibility of the tool's implementation and life cycle data, and other factors. Using such a design tool without independent assessment of the tool's output or establishing relevant history is discouraged, as it may prove to be a task as challenging as the development of the hardware for which the tool is proposed to be used.*

10. **Complete.** Document the tool assessment, justification for the assessment decisions, and if applicable, tool qualification data. Provide specific references to installation guides, user manuals and tool qualification data, as necessary to support the tool assignment and qualification.

11.4.2 Tool Assessment and Qualification Data

The tool assessment and qualification data should include:

1. Identify the tool, the process it supports and, when applicable, the following items:

   a. The rationale and results of the independent assessment per item 3 of Figure 11-1.

   b. The tool designation per item 4 of Figure 11-1.

   c. The tool's history when being used to satisfy item 5 of Figure 11-1. A discussion of the relevance of the previous tool usage versus the proposed usage of the tool should be included in the justification.

2. An unambiguous configuration definition to be used in tool qualification, in compliance with item 6 of Figure 11-1, and a justification for the applicability of the tested configuration if it differs from that actually used to design or verify the end hardware item.

3. Details of tool qualification, including the requirements used in testing, the test procedures, expected results, analysis procedures used to interpret and supplement the test results, and how independence is established.

4. The plan for qualifying a design tool, including the applicable procedures, and results for any activities identified in the plan.

5. The disposition of known tool errata, including workarounds, and, when applicable, problem reports generated as a result of tool qualification.

This Page Intentionally Left Blank

MEMBERSHIP

Special Committee 180/EUROCAE WG-46

Chairmen:

RTCA SC-180	Robert Clark	Honeywell, Inc.
RTCA SC-180	Lee Johnson	Rockwell Collins, Inc.
EUROCAE WG-46	Arnaud Demichelis	DGA/DCAE/STTE (FR)

Secretaries:

RTCA SC-180	Connie Beane	Federal Aviation Administration
EUROCAE WG-46	William Betts	Lucas Electronics (UK)
EUROCAE WG-46	Cleland Newton	DERA (UK)

RTCA Representative	EUROCAE Representative
Jack Cattilini	Francis Grimal
Jerry Bryant	Geoffrey Hunt
Rudy Ruana	

Joint Team 1 Co-Chairs:

Jacques Azum	Aerospatiale
Denis Mayfield	Boeing Commercial Airplane Group

Joint Team 2 Co-Chairs:

Ken Hunt	British Aerospace Airbus
Ted Parker	Honeywell, Inc.

Joint Team 3 Co-Chairs:

Brian Davis	Smiths Industries
David Richter	Rockwell Collins, Inc.

Joint Team 4 Co-Chairs:

David Austin	AlliedSignal
Francis Capecchi	Aerospatiale

Editorial Team:

Connie Beane	Federal Aviation Administration
Steven Beland	Boeing Commercial Airplane Group
Thierry Bickard	SNECMA
Francis Capecchi	Aerospatiale
Arnaud Demichelis	DGA/DCAE/STTE (FR)
Ken Hunt	British Aerospace Airbus
Lee Johnson	Rockwell Collins, Inc.
Thomas Neveling	DaimlerChrysler Aerospace Airbus
Cleland Newton	DERA (UK)
Ted Parker	Honeywell, Inc.
Dave Richter	Rockwell Collins, Inc.
William Struck	Federal Aviation Administration

Chris Wilkinson	Smiths Industries
Timothy Zimmerman	Cessna

MEMBERS

Ian Alderton	Ultra Electronics, Ltd.
Jozef van Baal	RLD
Barry Beaman	Woodward Governor Co.
Denys Bernard	Aerospatiale
Barbara BonJour	Boeing - Commercial Avionics Systems
Carmine Cifaldi	RAI
Christophe Conge	Dassault Aviation
Joe Costello	Rockwell Collins, Inc.
Rich Cuplin	Woodward Governor Co.
Dale Davidson	Honeywell, Inc.
Mike DeWalt	Certification Services, Inc.
William Donoghue	Pratt & Whitney
Cheryl Dorsey	Digital Flight
Joseph Eller	Liebherr-Aerospace
Brian Estep	Interstate Electronics Corp.
Bob Friday	AlliedSignal
Francois Gaffard	Intertechnique
Antoine Gautier	Dassault Aviation
John Glass	Smiths Industries
Bernard Gonzales	Bombardier Aerospace – Learjet, Inc.
Nathalie Goubert	STTE
Bill Greenleaf	Rockwell Collins, Inc.
Olivier Humez	Sextant Avionique
David Kirkland	Boeing Commercial Airplane Group
Tony Lambregts	Federal Aviation Administration
Dave Larkman	Boeing - Commercial Avionics Systems
Douglas Lee	Transport Canada
Michel Le Pimpec	Intertechnique
Brian Lucas	Proteus Corporation
Robert Maitan	Proteus Corporation
Joseph McHugh	ILC Data Device Corp.
Michael Miller	Honeywell, Inc.
Paul Miner	NASA, Langley Research
Ian Newton	GEC-Marconi Avionica
Jean Ofanowski	Dassault Aviation
Tom Olson	Rockwell Collins, Inc.
Steven Paasch	Certification Services, Inc.
Pascal Pampagnin	Aerospatiale
Chandrakant Patel	Litton Aero Products

Gerald Pilj	Bombardier Aerospace – Learjet, Inc.
Christian Pitot	Sextant Avionique
Benard Pro	AlliedSignal EAS
Misha Radich	Woodward Governor Co.
Leanna Rierson	Federal Aviation Administration
David Sandefur	Cessna
Paul Sapp	Universal Avionics Systems
Pete Saraceni	Federal Aviation Administration
Dennis Schmidt	Bombardier Aerospace – Learjet, Inc.
Kenneth Schmidt	Westinghouse ESG
Brian Shumaker	Proteus Corporation
Bruce Smith	Rockwell Collins, Inc.
Larry Smith	Honeywell, Inc.
Michael C. Smith	Ametek Aerospace
Pascal Thibault	Intertechnique
Laurie Thompson	Honeywell, Inc.
Jack Thornton	Rockwell Collins, Inc.
James Treacy	Federal Aviation Administration
Bertrand Voisin	Dassault Aviation
Kirk Walworth	Hamilton Sundstrand
Carl Ward	Lockheed Martin Aeronautical Systems
Brian Watkins	Bombardier Aerospace – Learjet, Inc.
Larry Yount	Honeywell, Inc

APPENDIX A

MODULATION OF HARDWARE LIFE CYCLE DATA BASED ON HARDWARE DESIGN ASSURANCE LEVEL

This appendix provides guidance for the modulation of the hardware design life cycle data based on the hardware design assurance level. It also provides guidance concerning the requirements for independence during the verification process.

Table A-1 identifies the data delivery classification and configuration management data control category for each data element. Refer to Table 7-1. There are two data delivery classification types defined:

1. **Submitted.** The data item should be submitted to the certification authority.

2. **Not Available.** The data item is not required.

All verification of Level A and B functions should be independent. Level C and lower functions do not require independent verification. Independence is needed only at the design hierarchy level at which the design is verified against the requirements. An equivalent means of independence, which addresses the issue of common mode failure, should be acceptable.

Independence is a means to address potential common mode errors that could occur when a designer verifies that the hardware item under development performs as designed, not as required. To address this concern, the responsibility for ensuring the verification process is consistent with demonstrating that the design requirements have been met should be performed with an individual, a process or a tool that is independent of the designer. There are many means of establishing independence and the verification plan should address the specific means to be used for a particular verification activity.

Some examples of acceptable means are:

1. Requirements or designs are reviewed by another individual.

2. Test cases or procedures are developed by another individual.

3. Test cases or procedures developed by the designer are reviewed by another individual.

4. An analysis performed by the designer is reviewed by another individual or a review team.

5. A different test is performed that confirms the results of testing by the designer, such as a test during flight test confirms a hardware item test or software verification tests, developed independently and performed on the target hardware item, confirm the results of testing by the designer.

6. Test or analysis results are verified by a tool.

Note 1: Often verification tests are automated and require only the "push of a key" to execute them. It is not the intent of independence to require someone other than the designer to execute the tests once they are evaluated or developed with independence. The results may still need to be reviewed independently to confirm proper procedures were followed and that the results verify that the requirements have been met.

Note 2: Organizational structure separation is not needed to achieve independence.

The circled numbers in Table A-1 refer to the notes following the table.

Table A-1 Hardware Life Cycle Data by Hardware Design Assurance Level and Hardware Control Category

Data Section	Hardware Life Cycle Data [1]	Objectives [2]	Submit	Level A	Level B	Level C	Level D
10.1	Hardware Plans						
10.1.1	Plan for Hardware Aspects of Certification	4.1(1,2,3,4)	S	HC1	HC1	HC1	HC1
10.1.2	Hardware Design Plan	4.1(1,2,3,4)		HC2	HC2	HC2	NA
10.1.3	Hardware Validation Plan [3] [4]	4.1(1,2,3,4); 6.1.1(1)		HC2	HC2	HC2	NA
10.1.4	Hardware Verification Plan	4.1(1,2,3,4); 6.2.1(1)	S	HC2	HC2	HC2	HC2
10.1.5	Hardware Configuration Management Plan	4.1(1,2,3,4); 7.1(3)		HC1	HC1	HC2	HC2
10.1.6	Hardware Process Assurance Plan	4.1(1,2,4); 8.1(1,2,3)		HC2	HC2	NA	NA
10.2	Hardware Design Standards						
10.2.1	Requirements Standards [3]	4.1(2)		HC2	HC2	NA	NA
10.2.2	Hardware Design Standards [3]	4.1(2)		HC2	HC2	NA	NA
10.2.3	Validation and Verification Standards [3]	4.1(2)		HC2	HC2	NA	NA
10.2.4	Hardware Archive Standards [3]	4.1(2);5.5.1(1); 7.1(1,2)		HC2	HC2	NA	NA
10.3	Hardware Design Data						
10.3.1	Hardware Requirements	5.1.1(1,2); 5.2.1(2); 5.3.1(2); 5.4.1(3); 5.5.1(1,2,3); 6.1.1(1,2); 6.2.1(1)		HC1	HC1	HC1	HC1
10.3.2	Hardware Design Representation Data						
10.3.2.1	Conceptual Design Data [3]	5.2.1(1)		HC2	HC2	NA	NA
10.3.2.2	Detailed Design Data	5.3.1(1); 5.4.1(2)		[5]	[5]	[5]	[5]
10.3.2.2.1	Top-Level Drawing	5.3.1(1); 5.4.1(2); 5.5.1(1)	S	HC1	HC1	HC1	HC1
10.3.2.2.2	Assembly Drawings	5.3.1(1); 5.4.1(2); 5.5.1(1)		HC1	HC1	HC1	HC1
10.3.2.2.3	Installation Control Drawings	5.4.1(2); 5.5.1(1)		HC1	HC1	HC1	HC1
10.3.2.2.4	Hardware/Software Interface Data [3]	5.3.1(1); 5.5.1(1)		HC1	HC1	HC1	HC1
10.4	Validation And Verification Data						
10.4.1	Hardware Traceability Data	6.1.1(1); 6.2.1(1,2)		HC2	HC2	HC2 [6]	HC2 [6]
10.4.2	Hardware Review and Analysis Procedures [3]	6.1.1(1,2); 6.2.1(1)		HC1	HC1	NA	NA
10.4.3	Hardware Review and Analysis Results [3]	6.1.1(1,2); 6.2.1(1)		HC2	HC2	HC2	HC2
10.4.4	Hardware Test Procedures [3]	6.1.1(1,2); 6.2.1(1)		HC1	HC1	HC2	HC2 [7]
10.4.5	Hardware Test Results [3]	6.1.1(1,2); 6.2.1(1)		HC2	HC2	HC2	HC2 [7]
10.5	Hardware Acceptance Test Criteria	5.5.1(3),6.2.1(3)		HC2	HC2	HC2	HC2
10.6	Problem Reports	5.1.1(3); 5.2.1(3); 5.3.1(3); 5.4.1(4); 5.5.1(4); 6.1.1(3); 6.2.1(4); 7.1(3)		HC2	HC2	HC2	HC2
10.7	Hardware Configuration Management Records	5.5.1(1); 7.1(1,2,3)		HC2	HC2	HC2	HC2
10.8	Hardware Process Assurance Records	7.1(2); 8.1(1,2,3)		HC2	HC2	HC2	NA
10.9	Hardware Accomplishment Summary	8.1(1,2,3)	S	HC1	HC1	HC1	HC1

① Data that should be submitted is indicated by an S in the Submit column. HC1 and HC2 data used for certification that need not be submitted should be available. Refer to Section 7.3.

② The objectives listed here are for reference only. Not all objectives may be applicable to all assurance levels.

③ If this data is used for certification, then its availability is shown in the table. This data is not always used for certification and may not be required.

④ This can be accomplished informally through the certification liaison process for Levels C and D. Documentation can be in the form of meeting minutes and or presentation material.

⑤ If the applicant references this data item in submitted data items, it should be available.

⑥ Only the traceability data from requirements to test is needed.

⑦ Test coverage of derived or lower hierarchical requirements is not needed.

APPENDIX B

DESIGN ASSURANCE CONSIDERATIONS FOR LEVEL A AND B FUNCTIONS

1.0 INTRODUCTION

The designer of hardware implementing Level A and Level B functions makes design decisions that may impact safety. As the design assurance level increases, the approach needed to verify that a given design meets its safety requirements may need overlapping, layered combinations of design assurance methods. It is up to the applicant to select one or more of these methods or propose another method that would provide design assurance.

This appendix provides the designer with guidance on how to perform and use an FFPA to develop a design assurance strategy as well as guidance on some specific methods that may be used for design assurance.

2.0 FUNCTIONAL FAILURE PATH ANALYSIS

An FFPA is a structured, top-down, iterative analysis. It identifies the specific portions of the design which implement the function; that is, the assemblies, components and elements associated with each path; and the associated failure modes and effects to be analyzed to determine that the hardware architecture and implementation complies with the safety requirements. FFPA also identifies those assemblies, components and elements of the design that implement the Level A and B functions.

An FFPA begins with the PSSA, which is used to identify system level FFPs that may be decomposed into and allocated to hardware FFPs.

The goal of an FFPA is to identify individual FFPs so that:

1. Hardware implementing Levels A and B functions can be addressed by an appropriate design assurance method described in this appendix or another advanced method acceptable to the certification authority.

2. Considerations of this appendix are optional for hardware implementing level C or lower level functions, that is, those functions that will be assured using only the guidance of Section 3 through Section 11 of this document.

Note: *Identification of separate FFPs for functions implemented in different technologies or offering different degrees of design visibility is often useful because the total hardware item's design assurance may be accomplished using multiple design assurance methods. The level of decomposition may vary for each FFP.*

Decomposition is performed using conventional top-down safety assessment techniques, such as fault tree analysis. The decomposition may be complemented using F-FMEA, dependency diagrams and common mode analysis for each successive level of decomposition. The level of decomposition may vary for each system level FFP depending on the design assurance strategy, corresponding implementation concept and the error mitigation methods being proposed for the hardware being designed. Decomposition progresses from:

system level FFPs	into	hardware level FFPs;
hardware level FFPs	into	circuit level FFPs;
circuit level FFPs	into	component level FFPs; and
component level FFPs	into	elemental level FFPs.

2.1 Functional Failure Path Analysis Method

The FFPA should be performed as follows:

1. For each Level A and Level B function, identify the function and its design assurance level based on the hardware requirements and system FHA for that function. The function may be formed as a collection of subfunctions, each having a corresponding set of derived requirements and an associated design assurance level. These subfunctions may be decomposed further as necessary.

2. For each Level A and Level B function, determine the means of implementing the function or the subfunctions and analyze the design assurance options. The assurance data available or expected to be available for the implementation of the function or subfunction should be complete and acceptable for the design assurance strategy or strategies chosen. If the assurance data available or expected to be available is complete, correct and acceptable, then no further decomposition is necessary.

3. For FFPs that are not Levels A or B, their interrelationships with the Level A or B FFPs should be evaluated using an F-FMEA, common mode analysis or dependency diagram to ensure that the Level A and B FFPs cannot be adversely impacted by the FFPs which are not Level A or B.

This assessment process is iterative. If there is no acceptable method of design assurance for a FFP, the decomposition and evaluation process is repeated or the architecture or implementation of the hardware function changed until an acceptable method of design assurance has been determined and acceptable assurance data is provided or can be provided for each Level A and Level B FFP.

Results of the FFPA and selected methods used for design assurance for the hardware are communicated to the aircraft systems process as described in Section 2.1 of this document. These results are used to examine and validate that the aircraft level assumptions, especially those related to multiple cross system usage of similar hardware items, are still valid.

2.2 Functional Failure Path Analysis Data

The FFPA data should:

1. Identify the anomalous behaviors and functional failures that have been delegated to the hardware item from the system level.

2. Identify the FFPs, the effects of their anomalous behavior or functional failure, and decomposition level in the design hierarchy to which the analysis was performed and the type and location of the acceptable assurance data that should be available.

3. Describe the relationship between FFPs to determine their independence and inter-dependencies on other FFPs and components. Such relationships may be described using qualitative FTA or other top-down analysis, common mode analysis, F-FMEA or dependency diagrams. The relationship descriptions should identify those inter-related paths and components and the inter-dependencies.

4. Trace between the FFPs and the hardware requirements and derived requirements.

3.0 DESIGN ASSURANCE METHODS FOR LEVEL A AND B FUNCTIONS

It is not the intent of this appendix to restrict the implementation of design assurance through the use of any current or future method. Methods discussed in this appendix may be used in satisfying one or more of the objectives of the processes described in Section 4 through Section 6 of this document.

3.1 Architectural Mitigation

Architectural design features, such as dissimilar implementation, redundancy, monitors, isolation, partitioning and command/authority limits, can be specifically employed to mitigate or contain the adverse effects of hardware design and implementation errors. As part of the PSSA, activities such as qualitative fault tree analysis and common mode analysis can provide assurance for determining the scope of architectural attributes needed to mitigate or contain the effects of hardware faults, failures, and design and implementation errors. More specifically, this approach should be applied in conjunction with the FFPA approach for hardware as described in Appendix B, Section 2, and should use the common mode analysis process to determine the applicability of particular mitigation strategies for coverage of hardware design and implementation errors. For example, redundancy usually helps mainly in the area of random faults or upsets, but redundancy can also be used effectively to mitigate design and implementation errors if their common mode aspects have been addressed.

3.1.1 Architectural Mitigation Method

Architectural mitigation is performed by identifying FFPs associated with a proposed hardware implementation, and then analyzing design options to identify and propose design features and strategies that mitigate the effects of these FFPs. The overall effects of a proposed architecture in regards to mitigating all relevant effects of the FFPs should be evaluated and addressed. Introduction of an architectural mitigation strategy also introduces some derived requirements against which its implementation should be verified. Specifically, the architectural features should protect against some or all of the adverse effects of the identified FFPs and should be assessed for introduction of additional failure paths, which should then be addressed by further architectural mitigation, or by another of the design assurance strategies described in this appendix.

3.1.2 Architectural Mitigation Resolution

The safety assessment process determines the acceptability of the architectural mitigation. The FFPA should first identify all the Level A and B hardware FFPs where architectural mitigation is to be used for credit, and should identify the methods to be used, and should determine the rationale for that mitigation. Adequacy is determined by assessing each function supporting the mitigation in the context of the overall architecture approach that may involve a more or less complex aggregate of architectural mitigation strategies.

The common mode analysis should address the potential for common mode errors in requirements, implementation, manufacturing and maintenance that could defeat the mitigation. The designer should also consider potential random failures of the hardware forming the architectural mitigation functions that may cause the mitigation to become unavailable. The probabilistic availability of the functions supporting the mitigation should be commensurate with the consequences of the loss of mitigation, which may result in the reduction of safety margins.

The overall approach should ensure that correct operation and acceptable independence between the necessary functions are achieved and maintained. Any special safeguards needed to eliminate, isolate or bound residual common mode effects should be identified and incorporated either in the form of additional architectural mitigation or other design assurance strategies defined in this appendix.

When the architecture definition is complete, hardware functions in Level A and B FFPs which are determined to be unmitigated, or inadequately mitigated, should be re-addressed using another design assurance methods from this appendix. For example, partial architectural mitigation of individual circuits and components can be used in conjunction with the safety specific analysis method when that analysis is used to identify and provide verification coverage for the unmitigated portions of the applicable circuits and components.

3.1.3 Architectural Mitigation Data

Documentation of architectural mitigation means, applied to protect levels A and B FFPs in hardware, should be provided in the forms of safety assessment data, safety requirements data and traceability data. The safety assessment data should be based on the assessment of hardware FFPs and common mode failure analysis specifically addressing the architectural mitigation aspects of the hardware design.

Architectural mitigation data should include:

1. Identification of the Level A and B hardware FFPs that are to be protected by architectural means.

2. Description of the architectural approach and validation rationale about coverage provided by that approach.

3. Rationale for common mode boundaries and common mode design aspects applicable to that architecture.

4. Identification of unmitigated and inadequately mitigated Level A and B FFPs to be addressed by other design assurance methods.

5. Requirements about the functional operation and necessary design attributes of the architectural mitigation mechanisms.

6. Mitigation mechanisms used to meet safety requirements that include software, such as software partitioning, safety monitors and dissimilar software. These mechanisms and safety software requirements should be provided to the system process and the software development process.

7. Conventional failure rate data and latent fault exposure assessment data for any hardware that performs the applicable architectural mitigation.

8. Traceability data linking safety requirements to the applicable safety assessment data and to the applicable design verification data.

3.2 Product Service Experience

Section 11.3 provides basic guidance on how to assess product service experience data for applicability for use in airborne hardware. For Level A and B functions that use previously developed hardware as part of the design, additional design assurance is necessary. This assurance can be provided in the following manner.

3.2.1 Product Service Experience Method

After completion of the assessment of Section 11.3, the FFPs that are implemented by the hardware under consideration should be analyzed with respect to any applicable service experience. The applicant or designer should identify the service experience data and establish that the service experience data demonstrates that the reused functionality of the hardware was sufficiently exercised during previous uses of the hardware.

3.2.2 Product Service Experience Resolution

When the service experience data analysis is complete, hardware functions in Level A and B FFPs that are determined to be not exercised, inadequately exercised or for which no service experience is available by in-service operation, should be addressed using another design assurance method or by the identification of additional verification that can be applied to exercise the functions.

3.2.3 Product Service Experience Data

Data of product service experience applied to protect Level A and B FFPs in hardware, should include:

1. The product service experience assessment data of Section 11.3.2.

2. Identification of the FFPs for which design assurance is provided by service experience and justification for the sufficiency of the service experience data.

3. Identification of the FFPs for which service experience data is insufficient and identification of test environments, test procedures, analyses and results used to complete the design assurance for the FFPs.

4. Identification of FFPs and operational conditions not demonstrated by the service experience that will require additional architectural mitigation or advanced verification method.

5. Traceability data as described in Section 10.4.1 showing the explicit relationship of the service experience data and verification that provides design assurance coverage of each FFP.

3.3 Advanced Verification Methods

Additional design assurance confidence may be achieved and evidence provided by the application of advanced verification methods, such as Elemental Analysis, Formal Methods, Safety-Specific Verification Analysis, or other applicant-proposed and certification authority-accepted methods.

The advanced verification methods of design assurance both use and extend the scope of the FFPA method presented in Appendix B, Section 2 The FFPA method is applied progressively at equipment-level, circuit-level, and component-level to determine the hardware implementation of the Level A and B FFPs. Data from the FFPA is then used to determine the proposed means of design assurance applicable to the hardware circuits, components and elements contained in those Level A and B FFPs.

These three methods are summarized here and described in the following sections.

1. **Elemental Analysis.** Elemental analysis provides a measurement of the completeness of the hardware verification from a bottom-up perspective. Every functional element within the FFP is identified and verified using verification test cases that meet the verification objectives of Section 6.1. The analysis may also identify areas of concern that need to be addressed by other appropriate means.

2. **Safety-Specific Analysis.** This strategy focuses on exposing and correcting the design errors that could adversely affect the hardware outputs from a system-safety perspective. Applicable safety sensitive portions of the hardware input space and output space are analytically determined. The sensitive portions of the hardware input space are stimulated, and the output space is observed not only for the safety-sensitive intended-function requirements verification, but also for anomalous behaviors. The methods of output space observation are identified in advance, by analysis that is accomplished using traditional safety analysis techniques.

3. **Formal Methods.** Formal Methods employ techniques from formal logic and discrete mathematics for the specification, design and verification of computer

systems. These techniques may be used to substantiate the reasoning employed in various processes of the hardware design life cycle.

Other advanced verification methods may be proposed by the applicant other than those described in this section.

3.3.1 Elemental Analysis

Elemental analysis may be used to show that FFPs are verified by associated verification test cases. Elemental analysis provides confidence and evidence that design errors are precluded by separating a complex implementation of the FFP into elements at the level that the designer generated it. This analysis method provides a measurement of the verification process to support the determination of verification coverage and completeness, and is most suited where the detailed design is visible and under configuration control. This would be the case in an ASIC or PLD, where the functions are addressed at the same design assurance level, or where functions of different design assurance levels are isolated or segregated. Every functional element of the applicable circuits or components is identified and verified for intended-function correctness using verification procedures that achieve the verification objectives of Section 6.1. Elemental analysis is generally applied to an entire component or an assembly without regard to the number of varied FFPs implemented in it, but may be applied to a portion of a component or assembly if a rationale can be provided for the isolation, independence or segregation of different FFPs.

Note: *When an elemental analysis is performed on a function implemented in a PLD, the programmed contents and the application of the PLD's features should be included, and the unprogrammed component may be addressed using a separate method, such as prior service experience.*

The analysis identifies areas of concern that need to be addressed by appropriate means. A verification process without such an analysis may leave some circuitry inadequately tested. Historically, such inadequacies have been due to shortcomings in requirements-based test procedures, unclear or incomplete hardware requirements, unused circuitry, initialization circuitry and library functions. This analysis examines verification of elements in the FFPs of concern and determines if the verification coverage related to each element is complete. Determination that verification coverage for elements is incomplete indicates a need for additional verification or appropriate activity.

The applicant should propose at what levels in the design hierarchy the elements are defined and how they are to be analyzed for verification coverage.

3.3.1.1 Elemental Analysis Method

The elemental analysis method begins by defining a set of criteria to be applied in the analysis in consideration of the hardware design assurance level, the hardware technology and visibility of the details of the implemented hardware.

The criteria should include:

1. Identification and a definition of the elements at an appropriate level of the hardware design.

2. The verification coverage to which each element should be verified.

These criteria are then applied to the analysis of verification activities to determine whether the verification coverage completion criteria will be achieved by the planned verification. If the criteria will not be achieved, then each element being examined should be exercised by an appropriate set of stimuli and cause appropriate observable effects on the signals being monitored in the test.

Note: *As this process examines the tests against the hardware itself, it can detect deficiencies in the test procedures. Addressing the test deficiencies would then provide additional confidence and evidence that the testing is sufficient, and the execution of new or amended test cases can then uncover errors in the hardware.*

3.3.1.1.1 Selecting Elemental Analysis Criteria

The elemental analysis criteria to be applied should be selected on a case-by-case basis depending on the hardware element type and complexity, and the identifiable functional operations of the element. The analysis may show either that all the low-level primitive blocks, such as counters, registers, multiplexers, adders, op amps and filters, have been adequately tested or that all groups of interconnected primitives have been adequately tested and achieve the verification coverage criteria. The analysis criteria of the test procedures should be derived based on an assessment of the functional operation of the element and its integration with other hardware elements in order to perform the next higher hierarchical level hardware function.

Note 1: *For example: if an element is a modulo-n counter used as a time delay, the test procedures may use appropriate equivalence-class selections of input data to verify that it counts when enabled, stops counting when disabled, counts at the correct rate, and reaches n and rolls-over at the specified time. It would not be necessary to show that the test procedures exercise the individual gates or flip-flops that collectively form the counter.*

As an example of using interconnected primitives as an element, an Arithmetic Logic Unit (ALU) may be constructed of primitives, such as registers, adders, and control logic. The ALU may be simulated to show that the primitives collectively form the ALU, but the verification procedures used in the elemental analysis should use appropriate equivalence-classes of input data to show that the ALU performs its functions.

The elements need not be defined at a level of the design below that specified by the designer of the hardware. Gate-level analysis may be appropriate only if the design is explicitly expressed as gates for combinatorial logic or state machine control.

Note 2: *Analyzing the implementation below the level of that specified by the designer, such as at the gate or transistor level, is not necessary as it would be analogous to analyzing software at the assembly language or binary pattern level. These lower abstraction levels are implicitly addressed by performing the elemental analysis on verification tests performed on the hardware, or on post-layout simulations successfully assessed, and if necessary, qualified as verification tools per Section 11.4.*

An ASIC or PLD may contain proprietary library functions that may not provide visibility of their internal design and therefore would not lend themselves to manual analysis. Library functions may be treated as COTS elements in the elemental analysis, with the COTS hardware aspects addressed as defined in Section 11.2 and Appendix B, Section 2.2. Verification of the application of the library function should show that it is consistent with its specification or description provided by the library manufacturer and the tests should be executed in an environment that allows the test results to be observed.

Note 3: *The intent is not to discourage the use of design libraries in favor of building new functions; the practical use of design libraries is encouraged to minimize further opportunities for introducing errors into the hardware.*

For ASICs or PLDs synthesized from a high level description in an HDL, the analysis criteria may be based on the high-level behavioral language code representing the hardware. However, since implementations synthesized from HDL representations may include parallel logic structures and non-sequential temporal aspects, the synthesized output should be included in the analysis completion determination. The synthesizer should be assessed as well.

3.3.1.1.2 Performing the Elemental Analysis

Elemental analysis should use the requirements-based verification tests performed in one or more of the following test environments:

1. Tests with the circuitry implementing the functional path installed in the target assembly.

2. Tests performed on a standalone prototype. Such tests are typical for an ASIC or PLD.

3. Manufacturing acceptance tests.

   Note: *Since manufacturing tests often are not based on the requirements, manufacturing acceptance tests may be restricted in their application to elemental analysis.*

4. A post-layout simulation, typically for an ASIC or PLD, that has been assessed and, if necessary, qualified for use as a verification tool as described in Section 11.4.

An elemental analysis itself may be performed using a simulation to measure the completeness achieved, provided that the test procedures to be analyzed can be related to the elemental analysis criteria being applied and are those used for hardware functional verification credit toward the objectives in Section 6. If the test procedures analyzed are derived from an in-circuit test of hardware or standalone prototype and are examined using a simulation, the test stimuli and expected results may be translated for the simulator provided that the translation process is checked for accuracy as a part of the elemental analysis. A simulator used to perform the elemental analysis should be shown to be able to correctly determine whether each type of element included in the implementation has met the analysis criteria.

3.3.1.2 Elemental Analysis Results Resolution

Elemental analysis may reveal hardware elements not verified, indicating either a need for additional verification process activities or perhaps a need to remove the untested element or mitigate any anomalous behavior that could result by architectural means. Untested hardware elements may be the result of:

1. **Shortcomings in verification test cases or procedures.** Shortcomings may arise if the test cases simply do not test the elements in the hardware item in compliance with the criteria in Appendix B, Section 3.3.1.1. They may also arise if there are "don't cares" in the functional requirements but the hardware item was appropriately designed to produce repeatable responses. Under these circumstances, the test procedures and cases should be supplemented or changed. Furthermore, the assertion of the test's ability to verify its respective requirements should be reviewed.

2. **Inadequacies in requirements.** The requirements should be modified or additional derived requirements identified. Additional verification tests should then be developed for the new or revised requirements, executed and analyzed.

3. **Unused functions.** The hardware item may contain functions that are not used in its target circuit application, such as unused subfunctions within a library function or test structures used only for component-level acceptance tests. Such functions should either be shown to be isolated from the other used functions or shown to present no potential anomalous behavior that could have an adverse effect on safety. This could possibly be achieved by showing that the unused elements are positively deactivated either within the hardware or when installed. If the unused functions are to be used in some future application, the elemental analysis deficiency may be revisited at that time provided that such functions are identified as not being fully verified.

4. **Element of no safety consequence.** The consequence of anomalous behavior of the element can be bound and shown by analysis to not cause an adverse safety effect to the airplane or its occupants. These items should be resolved by recording the analysis bounding the consequence of anomalous behavior of the element.

3.3.1.3 Elemental Analysis Life Cycle Data Output

The elemental analysis life cycle data output should:

1. Identify the FFPs to be addressed by elemental analysis, and propose at what levels in the design hierarchy the elements are defined and how they are to be analyzed for

verification adequacy, which are parts of the verification coverage completion criteria. This should be included in the PHAC or hardware verification plan.

2. Describe the methods and identify the FFPs addressed in the analysis and the levels in the design hierarchy at which the analysis was performed.

3. Ensure that the traceability data, as described in Section 10.4.1 shows the explicit relationship of the verification procedures to the elements in the elemental analysis.

4. Identify the verification test cases and requirements added or modified as a result of the elemental analysis.

5. State the level of the verification completeness achieved for the FFPs addressed by elemental analysis, including identification of the analysis discrepancies not resolved by modification to verification tests or requirements and the rationale for acceptability.

3.3.2 Safety-Specific Analysis

Where applied, the safety-specific analysis method extends the hardware FFPA concept by performing a more in-depth analysis of the selected circuits and components. The extended FFPA is used to both derive and validate safety-specific requirements about internal operations of those circuits and components. These derived safety requirements are then addressed by the verification tests as discussed below.

Safety-specific analysis is based on the concept that a potentially latent design error can affect a hardware item's output only when specific input stimuli expose it. Therefore, to properly stimulate and expose the safety errors of concern, the subset of input cases for which safe operation is necessary is identified and then appropriate equivalence classes from that subset are included in the verification tests. During execution of these test cases, the item's outputs are evaluated for absence of specific anomalous behaviors that could result in unsafe output conditions. The safety-specific analysis is used to bound the set of input conditions to be applied in the verification test cases so that a potentially infinite set of input test cases do not have to be addressed.

Note: The implementation may also bound the input set and conditions so that it is not possible or is adequately improbable that the implementation would allow an input outside the limits tested.

The safety-specific analysis method can also be used to determine the unmitigated aspects of circuit and component functions in which partial architectural mitigation is applicable. In this case, the additional safety-specific analysis can be a useful and effective method to determine what additional design assurance is needed to complete the safety coverage.

The safety–specific analysis method is equally applicable to either COTS hardware or custom circuits and components because it is able to use user guide data about those circuit and components instead of detailed internal design data. By combining the user guide data with this more detailed application of the FFPA method, the safety-specific analysis is able to successfully determine the safety-sensitive aspects of circuit and component usage and the associated internal FFPs where design error removal emphasis is needed. This information can then be used to successfully derive circuit and component verification tests which, when completed, maximize the likelihood that the verification process has exposed and corrected, mitigated, or provided work-arounds for those circuit and component design errors which could adversely affect the hardware from a system-safety perspective.

3.3.2.1 Safety-Specific Analysis Method

Once the circuits and components which are to be addressed using the safety-specific analysis method of design assurance are selected, then an additional FFPA is performed to examine them in greater detail. This analysis determines more specifically which circuit and component functions contribute to the already identified Level A and B functions that use those circuits and components. This is accomplished by examining each applicable circuit and component, case-by-case, at its functional boundaries, considering the actual functional usage of that circuit or component to perform the higher level hardware functions contained in the identified Level A and B FFPs.

Note: *Sufficient information may be available in circuit and component user's guide data that a user can successfully use the functions of that circuit or component to perform higher level hardware functions. If sufficient information is available about the circuit's or component's internal functioning, it should also be adequate to make this assessment. If sufficient information is not available, this assessment cannot be done, and another method should be used instead or in conjunction with this method.*

After the relevant safety-sensitive functions of the circuits and components have been identified based on the actual usage of those circuits and components, the next step is an even more detailed functional analysis. This analysis should determine the specific safety-sensitive and unmitigated attributes of those circuit and component functions that are to be addressed in more detail by the safety-specific verification conditions. These verification conditions should be derived and validated by using F-FMEA techniques to determine the specific functional attributes that are safety-sensitive and further to determine any specific anomalous behavior of those functions that would constitute a Level A and B FFP within the circuit or component.

Derived verification conditions obtained via the above safety-specific analysis activities are then used in conjunction with the following guidance to complete the safety-specific analysis criteria for verification of circuit and components contained in Level A and B FFPs. Guidance includes:

1. Identify the relevant input space of the functions. Determine the associated output space pass/fail criteria, based on the identified safety-sensitive functional attributes and anomalous behaviors, and develop the equivalence-classes that will provide the necessary coverage of the relevant input space.

2. Identify relevant observable detection means, and input space stimulation means for each considered function.

Note: Special tools and implementation features may be used to ensure observe-ability and testability.

3. Specify the test environments that address verification of potential error sources and interdependencies.

Note: Component-level functions should be tested at the highest integration level feasible. Testing at higher levels of integration usually provides the best coverage of error-sources, such as upset, interdependencies and potential cross-functional interactions.

Tests should be developed using equivalence-classes. Testing should address key logic decisions, arithmetic, timing, state transitions and real-time attributes.

3.3.2.2 Safety-Specific Analysis Resolution

The safety-specific verification completion criteria should be established by completion of the safety-specific analysis for all the applicable circuits and components. Any deficiencies found by that analysis or by the verification itself should be resolved by one of the following methods:

1. Change the design to correct the error.

2. Add architectural mitigation, which resolves the error by removing it from the relevant FFP.

3. Add appropriate tests.

3.3.2.3 Safety-Specific Analysis Data

Documentation of safety-specific analysis, when applied to circuits and components in Level A and B FFPs, should be provided in the form of safety assessment data, safety requirements data, verification procedures and results, and traceability data. The verification procedures should be traceable to the safety requirements, and to the safety-specific analysis. Safety-specific analysis data should include:

1. Identification of the circuit and components which are to be addressed by the safety-specific analysis method.

2. Identification of the Level A and B FFPs in which each of those circuits and components reside.

3. Identification of partial architectural mitigation applicable to circuits and components where design assurance completion is to be provided by the safety-specific analysis method.

4. For each applicable circuit and component, identification of safety sensitive functions.

5. For each identified safety-sensitive function, identification of safety-sensitive attributes and anomalous behaviors of concern.

6. Verification conditions addressing the applicable circuits, components, internal functions, functional attributes and anomalous behaviors.

7. Verification conditions addressing input dependencies and output space behaviors to be verified.

8. Verification procedures and results.

9. Traceability data linking verification procedures and hardware safety verification conditions to safety-specific hardware analysis data.

3.3.3 Formal Methods

The term formal methods refers to the use of techniques from logic and discrete mathematics in the specification, design and construction of computer systems.

Note: *The material in this section is derived from "Formal Methods Specification and Analysis Guidebook for the Verification of Software and Computer Systems, Volume II: A Practitioner's Companion," May 1997, NASA-GB-001-97. A more detailed presentation of the application of formal methods, illustrated with a worked example, can be found there.*

Applications of formal methods fall into two broad categories, descriptive and deductive. Descriptive methods employ formal specification languages, which provide for clear, unambiguous descriptions of requirements and other design artifacts. Deductive methods rely on a discipline that requires the explicit enumeration of all assumptions and reasoning steps. In addition, each reasoning step must be an instance of a small number of allowed rules of inference. The most rigorous formal methods apply these techniques to substantiate the reasoning used to justify the requirements, or other aspects of the design or implementation of a complex or critical system. The purpose of formal methods is to reduce reliance on human intuition and judgment in evaluating arguments. That is, deductive formal methods reduce the acceptability of an argument to a *calculation* that can, in principle, be checked by a tool, thereby replacing the inherent subjectivity of the review process with a repeatable exercise.

There are several areas where application of formal methods provides additional assurance in the design process. Although formal methods are applicable throughout the design process, increases in design assurance may be obtained by targeted application. The following list highlights some of the possibilities:

1. Formal methods may be applied at different stages of the development life cycle. Generally, applications of formal methods are most effective at the early stages of the life cycle, specifically during requirements capture and high-level design.

2. Formal methods may be applied to the entire design or they may be targeted to specific components. The FFPA is used to determine which FFPs to analyze with formal methods. Protocols dealing with complex concurrent communication and hardware implementing fault-tolerant functions may be effectively analyzed with formal methods.

3. Formal methods may be applied to verify system functionality or they may be used to establish specific properties. Although formal methods have traditionally been associated with "proof-of-correctness," that is, ensuring that a component meets its functional specification, they can also be applied to only the most important properties. Often, it is more important to confirm that a design does not exhibit certain undesirable properties, rather than to prove that it has full functionality.

Practical application of formal methods typically requires tool support. Tools used should be assessed and, if necessary, qualified as described in Section 11.4.

3.3.3.1 The Methodology of Formal Methods

The application of formal methods begins by expressing the requirements using a formal language. The requirement specification serves an important descriptive function. It provides a basis for documenting, communicating and prototyping the behavior and properties of a system using an unambiguous notation. In addition, the requirements specification serves as a basis for calculating or formally predicting system behavior. A formal model of the component to be analyzed is constructed using a formal language. The model is analyzed with respect to the formal statement of requirements using the rules of the selected formal logic. The characteristics of the model are determined by the style of formal analysis to be performed.

The level of detail in the component model is determined by the goal of the chosen formal analysis technique. Some approaches are tailored to finding design errors that may have eluded testing, while other approaches seek to guarantee the absence of certain classes of design errors.

1. **Error-Detection.** The most common formal technique for error detection is called model checking. Here the requirements are expressed as formula in a decidable temporal logic. The model of the component is an abstract state machine designed so that the property to be tested is preserved. The proof procedure is automatic. A failed proof attempt indicates a design error in the modeled component. The result of failed proof is a sequence of input stimuli that demonstrate specifically how the component does not satisfy the stated requirement.

2. **Error Preclusion.** Formal methods targeted to prevention of errors are generally based upon an expressive specification language with a supporting proof theory. With the increased expressiveness, more complicated requirements may be stated and more detailed models of the component may be constructed. However, the proof procedure may only be partially automated. An appropriate level of detail for the component model may be a synthesizable HDL description. In some cases, the same model may be used both for simulation and formal analysis. A completed proof is evidence that the component is logically correct with respect to the stated requirements for the analyzed input space.

3.3.3.2 Formal Methods Resolution

There are three possible outcomes of a deductive formal analysis:

1. If the proof attempt is successful, the verification activity is complete. The level of design assurance depends upon the fidelity of the models employed. For example, if the model of the hardware item corresponds to a detailed design, the proof provides assurance of functional correctness equivalent to that of exhaustive testing.

2. In some cases, a failed proof results in an explicit counter-example; that is, it identifies a test scenario to illustrate specifically how the design does not meet the stated requirements. This may indicate either a deficiency in the design or a deficiency in the requirements. Such deficiencies may be resolved by correcting the design, revising the requirements, shown to not be a physically realizable condition or using another method. All counter-examples should be identified so that they can be resolved. Changes to the design or requirements need to be reflected back to the appropriate process.

   a. After a design or requirement has been modified to address a deficiency identified by a failed proof attempt, the proof should be attempted again to confirm that the modification has successfully addressed the identified problems. This cycle is repeated until a successful proof is achieved.

   b. In cases where a counter-example is considered resolved without requirement or design changes but the tool identifies only one counter-example, that is, the resolved counter-example, the process should be modified so that it can identify all other counter-examples.

3. The most difficult case to resolve is when a proof cannot be produced and a counter-example cannot be identified. One possible option is to revise the design in order to simplify the verification effort. Alternatively, the verification activity may be decomposed with a clear delineation between the cases addressed by proof and those cases where the requirement needs to be addressed by some other means. Changes to the design and derived requirements should be reflected back to the FFPA.

3.3.3.3 Formal Methods Data

The data developed during the application of formal methods includes:

1. Description of the specific formal methods approach to be used and the components or FFPs to which formal methods will be applied.

2. Formal statement of requirements.

3. Formal models of the component.

4. Proof, or sufficiently detailed script to generate proof, relating the models of the component to the formal statement of requirements and including correlation in the traceability data.

5. Identification of tools employed and tool assessment results.

6. Identification of the verification test cases and requirements added or modified as a result of the analysis.

7. Statement of the level of the verification completeness achieved for the FFPs addressed by analysis. Include a list of the analysis discrepancies not resolved by modification to verification test cases or requirements and their rationale for acceptability of the discrepancies.

APPENDIX C

GLOSSARY OF TERMS

These definitions are provided for the terms as used in this document. If a term is not defined in this appendix, it may be defined in the associated body of text.

Acceptance - Acknowledgment by the certification authority that a submittal of data, argument or claim of equivalence satisfies applicable requirements.

Airworthiness - The condition of an item, which can be an aircraft, aircraft system or component, in which that item operates in a safe manner to accomplish its intended function.

Analysis - A process of mathematical or other logical reasoning that leads from stated premises to the conclusion concerning specific capabilities of equipment or hardware item and its adequacy for a particular application.

Anomalous Behavior - Behavior that is inconsistent with specified requirements.

Applicant - A person or organization seeking approval from the certification authority.

Application Specific Integrated Circuit (ASIC) - Integrated Circuits which are developed to implement a function, including, but not limited to: gate arrays, standard cell and full custom components encompassing linear, digital and mixed mode technologies.

Approval - The act or instance of expressing a favorable opinion or giving formal or official sanction.

Assembly - A number of components or any combination thereof, joined together to perform a specific function.

Assessment - An evaluation based upon engineering judgment.

Assumptions - Statements or principles offered without proof.

Assurance - The result of planned and systematic actions necessary to provide adequate confidence and evidence that a product or process satisfies given requirements.

Availability - Probability that an item or function is in an operable state.

Baseline - An identified and approved configuration that thereafter serves as the basis for further design, and that is changed only through change control procedures.

Certification - Legal recognition by the certification authority that a product, service, organization or person complies with the requirements. Such certification comprises the activity of technically checking the product, service, organization or person and the formal recognition of compliance with the applicable requirements by issue of a certificate, license, approval or other documents as are required by national laws and procedures. In particular, certification of a product involves:

a. The process of assessing the design of a product to ensure that it complies with a set of standards applicable to that type of product so as to demonstrate an acceptable level of safety.

b. The process of assessing an individual product to ensure that it conforms with the certified type design.

c. The issuance of a certificate required by national laws to declare that compliance or conformity has been found with standards in accordance with the above two items.

Certification Authority - The organization or person responsible within the state or country concerned with the certification of compliance with the requirements.

Note: *A matter concerned with aircraft, engine or propeller type certification or with equipment approval would usually be addressed by the certification authority; matters concerned with continuing airworthiness might be addressed by what would be referred to as the airworthiness authority.*

Certification Basis - Defined by the Certification Authority in consultation with the Applicant, as the particular certification requirements, together with any special conditions which may supplement the published regulations, that become the basis for certification of the aircraft, engine, or propeller.

Certification Credit - Acceptance by the Certification Authority that a process, product or demonstration satisfies a certification requirement.

Change Control - (1) The process of recording, evaluating, approving or disapproving, and coordinating changes to configuration items after formal establishment of their configuration identity, or to a baseline after its establishment. (2) The systematic evaluation, coordination, approval or disapproval and implementation of approved changes in a configuration of a configuration item after formal establishment of its configuration identity or to baseline after its establishment.

Commercial Off-The-Shelf (COTS) Component - Component, integrated circuit or subsystem developed by a supplier for multiple customers, whose design and configuration is controlled by the supplier's or an industry specification.

Note: *Examples of COTS components may include resistors, capacitors, microprocessors, unprogrammed Field Programmable Gate Array and Erasable Programmable Logic Devices, other integrated circuit types and their implementable models, printed wiring assemblies and complete LRUs which are typically available from a supplier as a catalog item.*

Common Mode - Event which causes anomalous behavior of two or more items, subitems or functions.

Complex Hardware Item - All items that are not simple are considered to be 'complex'. See definition of Simple Hardware Item.

Compliance - Successful performance of all mandatory activities, agreement between the expected or specified result, and the actual result.

Component - A self-contained part, combination of parts, subassembly or unit that performs a distinct function of a system.

Component De-rating - This is a design method which increases the operational margins of components by imposing modified component usage limitations which are more restrictive than the usual or manufacturer's component operational ratings.

Concurrent Engineering - A process whereby multiple disciplines participate in the hardware design process in order to ensure that the unique requirements of each discipline are considered.

Configuration - A list of Configuration Items that completely defines an implementation of a function.

Configuration Identification -The process of defining and designating a Configuration Item.

Configuration Identity - The unique name given to a configuration item or to a configuration as the result of Configuration Identification.

Configuration Item - One or more components, tools or data items treated as a unit for configuration management purposes.

Configuration Management - (1) The process of Configuration Identification, and the control of issues and changes of Configuration Identities. (2) A discipline applying technical and administrative direction and surveillance to identify and record the functional and physical characteristics of a configuration item, control changes to those characteristics, and record and report change control processing and implementation status.

Conformance - Established as correct with reference to a standard, specification or drawing.

Conformity - Agreement of physical realization of the hardware item with the defining documents.

Coverage Analysis - The process of determining the degree to which a proposed hardware verification process activity satisfies its objective.

Defect - Any non-conformance of a characteristic with specified requirements.

Derived Requirement - Additional requirement resulting from the hardware design processes, which may not be directly traceable to higher level requirements.

Design Assurance – All of those planned and systematic actions used to substantiate, at an adequate level of confidence, that design errors have been identified and corrected such that the hardware satisfies the application certification basis.

Design Margin Analysis - The process of determining that the sum effect of various hardware component design margins provides a product which meets or exceeds its performance requirements as well as requirements for producibility and service.

Design Process - The process of creating a hardware item from a set of requirements using the following set of processes: requirements capture, conceptual design, detailed design, implementation and production transition.

Design Tools - Tools whose output is part of hardware design and thus can introduce errors. For example, an ASIC router or a tool that creates a board or chip layout based on a schematic or other detailed requirement.

Equivalence Class – The partitions of the input space of a function such that a test of a representative value of the class is equivalent to a test of other values of the class.

Error - A mistake in requirements, design or implementation.

Exposure Time - The period of time between when a hardware item was last known to be operating properly and when it will be known to be operating properly again.

Failure - The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered.

Failure Condition - The effect on the aircraft and its occupants both direct and consequential, caused or contributed to by one or more failures, considering relevant adverse operational and environmental conditions.

Failure Effect - (1) A description of the operation of an item as the result of a failure; (2) the consequences a failure mode has on the operation, function, or status of a system or an item.

Failure Mode - The way in which the failure of an item occurs.

Failure Rate - The total number of failures within an item population, divided by the total number of item power-on hours under stated conditions.

Fault - (1) A manifestation of a flaw in hardware due to an error or random event. A fault, if it occurs, may cause a failure. (2) An undesired anomaly in an item.

First Article - A unit submitted for inspection to verify the production drawings, tools and procedures.

First Article Inspection - A Process Assurance inspection that verifies that the hardware “as-built” conforms to the manufacturing process documentation. Performed on production hardware items representing first-off-the-line configuration as a precondition for production approval.

Functional Defects - Defects which cause hardware functions to operate incorrectly, even though a hardware physical failure has not occurred. Resultant incorrect hardware operation in turn may cause dependent software functions to operate incorrectly.

Functional Failure Path - The specific set of interdependent circuits that could cause a particular anomalous behavior in the hardware that implements the function or in the hardware that is dependent upon the function.

Functional Path - The specific set of interdependent circuits that implement a function.

Glitch – An input transition or voltage spike that occurs in a time period that is shorter than the delay through the affected logic that can propagate to the output.

Guidance - Advice or counseling for complying with certification requirements.

Hardware Design Life Cycle Process - One of the set of design or supporting processes determined by an organization to be sufficient for the design of a hardware item.

Hardware Description Language - HDL is used in this document to represent all of the Hardware Description Languages, including “Verilog HDL”, Very High Speed Integrated Circuit Hardware Description Language and Analog Hardware Description Language.

Hardware Item - An item that has physical being. This generally refers to LRUs, circuit board assemblies, power supplies and components.

Hardware Partitioning - A method for enhancing reliability and safety by physical separation and isolation of the hardware that is implementing the functions, including redundancy, to prevent failure effects due to common faults.

Hardware/Software Integration - The joining of hardware and software to implement an application or function.

Independence - Separation of responsibilities which ensures the accomplishment of objective evaluation. Refers to intellectual independence, such as another individual, and not departmental or company independence.

1. For verification, independence is achieved by evaluation of the technical correctness of the data by means, either someone or something, other than those used to produce the data.

2. For process assurance, independence is achieved by evaluation of process compliance by means, either someone or something, other than those used to perform the process.

Implementation - The act of generating a physical reality from a specification.

Inspection - The examination and testing of supplies and services, including when appropriate, raw materials, components, intermediate assemblies and services, to determine whether they conform to specified requirements.

Integrated Circuit - A circuit consisting of elements inseparably associated and formed in-situ on or within a single substrate to perform an electronic circuit function.

Integrity - Attribute of an item indicating that it can be relied upon to perform the intended function.

Item -A general term used to refer to a subject hardware component, system or software.

Life Cycle - the period of time between starting the design or modification of a hardware item and completing the design or modification up as far as transition to production.

Note: *In this document, unless defined otherwise in the text, this means "Hardware Design Life cycle"*

Maintainability - A characteristic of design and installation which is expressed as the probability that an item will be retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources.

Malfunction -The occurrence of a condition whereby the operation is outside specified limits.

Manufacturability - Product design features which facilitate economic mass production by optimizing materials and manufacturing tools and by employing design techniques which minimize the impact of component variations on functionality.

Means of Compliance - The methods to be used by the applicant to satisfy the requirements stated in the certification basis for an aircraft or engine. Examples include statements, drawings, analyses, calculations, testing, simulation, inspection and environmental qualification. Advisory material issued by the certification authority is used if appropriate.

Monitoring - (1) **Safety.** Functionality within a system that is designed to detect anomalous behavior of that system. (2) **Process Assurance.** The act of witnessing or inspecting selected instances of test, inspection, or other activity, or records of those activities, to assure that the activity is under control and that the reported results are representative of the expected results. Monitoring is usually associated with activities done over an extended period of time where 100% witnessing is impractical or unnecessary. Monitoring permits authentication that the claimed activity was performed as planned.

Over-stress defects - Defects which either cause a component to exceed rated design limits or result from over-stress encountered during the hardware design life cycle.

Part Number - A set of numbers, letters or other characters used to identify a configuration item, a configuration identity.

Planning Process – A process to define and coordinate the activities of the hardware design and support processes.

Preliminary System Safety Assessment – A systematic evaluation of a proposed system architecture and its implementation, based on the functional hazard assessment and failure condition classification, to determine safety requirements for all items in the architecture.

Note: A Preliminary Systems Safety Assessment applies to the system under development. It is used to direct further safety analysis activity required to complete the final system safety assessment.

Process - A set of interrelated activities performed to produce a prescribed output or product.

Process Assurance – The objective of process assurance is to ensure that plans are followed, hardware design life cycle process objectives are met and activities have been completed.

Product - Hardware, software, item or system generated in response to a defined set of requirements.

Product Service Experience - A period of time during which the hardware is operated within a known environment and during which successive failures are recorded.

Production -Manufacture of product by a documented and controlled sequence of processes.

Programmable Logic Device (PLD) - A component that is purchased as an electronic component and altered to perform an application specific function. PLDs include, but are not limited to, Programmable Array Logic components, Programmable Logic Array components, General Array Logic components, Field Programmable Gate Array components and Erasable Programmable Logic Devices.

Prototype - A pre-production hardware item that is fully representative of the final product using approved components and suitable for complete evaluation of form, design and performance.

Release – The act of formally placing the data of a hardware item under configuration control.

Reliability - The probability that an item will perform its intended function for a specified interval under stated conditions.

Reliability Defects - Defects that cause hardware to fail at an excessive rate when subjected to stress conditions not exceeding rated design limits. Both over-stress defects and reliability defects may be manifested as excessive random failure rate, excessive infant mortality or excessive wear-out rate.

Requirement - An identifiable element of a specification that is verifiable.

Reverse Engineering - Re-implementation of a hardware item by study of its construction, function and performance within a particular environment.

Review - Qualitative evaluation to assess the plans, requirements, design data, design concept or design implementation to demonstrate to a high degree of confidence that the requirements have been or will be met.

Risk - The combination of the frequency and the consequence of a specified hazardous state.

Robustness Defects - Defects that cause hardware to fail or operate incorrectly when subjected to stress conditions and service life not exceeding design limits. Results of these defects may include susceptibility to environmental stress and instability over service life.

Safety - The state in which risk is lower than the boundary risk. The boundary risk is the upper limit of the acceptable risk. It is specific for a technical process or state. The risk is defined by the rate or probability of occurrence and the expected damage or injury.

Similarity - Applicable to systems comparable in characteristics and usage to systems used on an airplane previously certificated by the applicant. It is further assumed that there are no parts of the subject system are more at risk due to environment or installation and that operational stresses are no more severe than on the analogous system.

Simple Hardware Item - A hardware item is considered simple if a comprehensive combination of deterministic tests and analyses can ensure correct functional performance under all foreseeable operating conditions with no anomalous behavior.

Simulator - A device, computer program or system used during hardware verification, that accepts the same inputs and produces the same output as a given system.

Software - Computer programs and, possibly associated documentation and data pertaining to the operation of a computer system.

Specification - A collection of requirements that, when taken together, constitute the criteria which define the functions and attributes of an item.

Standard - A rule or basis of comparison used to provide both guidance in and assessment of a given activity or the content of a specified data item.

Structure - A specified arrangement or interrelation of parts to form a whole.

Supporting Process – A process used to support the design process consisting of one of the following set of processes: validation, verification, configuration management, process assurance and certification liaison.

System Architecture - The structure of the hardware and the software selected to implement the system requirements.

System - A collection of hardware and software components organized to accomplish a specific function or set of functions.

System Safety Assessment (SSA) - An ongoing, systematic, comprehensive evaluation of the proposed system to show that relevant safety requirements are satisfied.

Test - A quantitative procedure to prove performance using stated objective criteria with pass/fail results.

- Hardware Item. To determine its performance characteristics while functioning under controlled conditions.
- Electronic digital computation. To ascertain the state or condition of an element, component, program, etc.
- Sometimes used as a general term to include both check and diagnostic procedures.
- Loosely, same as check.
- Is an element of inspection and generally denotes the determination by technical means of the properties of elements of supplies, or comments thereof, including functional operation, and involves the application of established scientific principles and procedures.

Testability - (1) The ability to test a hardware item sufficiently to guarantee that all possible states of the hardware item performs to its specification. (2) The ease with which a hardware item can be tested to provide evidence of compliance with its requirements.

Testing - The process of verifying the performance of a hardware item.

Test Procedure - Detailed instructions for controlling the conditions for executing a given set of tests.

Tool Assessment - A set of activities to assess the tools used in the design and verification of the hardware item to provide confidence that the tool is capable of performing its functions correctly consistent with the design assurance level of the functions to be performed by the hardware item.

Tool Qualification - The process necessary to obtain certification credit for a tool within the context of a specific airborne system.

Traceability - An identifiable association between hardware items or processes, such as between a requirement and the source of the requirement or between a verification method and its base requirement.

Upset - Interference caused by external events, such as lightning or other environmental events.

Validation - The process of determining that the requirements are the correct requirements and that they are complete.

Verification - The evaluation of an implementation of requirements to determine that they have been met.

Verification Tool - Tools used to ensure performance against predetermined standards or requirements. These tools do not introduce errors, but may fail to detect them. For example, an analog or digital circuit simulator or an automated test that measures actual circuit performance.

APPENDIX D

ACRONYMS

ALU	Arithmetic Logic Unit
ARP	Aerospace Recommended Practice
ASIC	Application Specific Integrated Circuit
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
JAR	Joint Aviation Requirements
LRU	Line Replaceable Unit
PHAC	Plan for Hardware Aspects of Certification
PLD	Programmable Logic Device
PSSA	Preliminary System Safety Assessment
RTCA	RTCA, Inc.
SAE	Society of Automotive Engineers
SC	Special Committee
SSA	System Safety Assessment
WG	Working Group