Certification Authorities Software Team (CAST)

Position Paper CAST-30

Simple Electronic Hardware and RTCA Document DO-254 and EUROCAE Document ED-80, *Design Assurance Guidance for Airborne Electronic Hardware*

COMPLETED August 2007

(Rev 0)

NOTE: **This position paper has been coordinated among representatives from certification authorities in North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

Simple Electronic Hardware and RTCA Document DO-254 and EUROCAE Document ED-80, *Design Assurance Guidance for Airborne Electronic Hardware*

1.0 Purpose

This CAST paper provides clarification to the guidance in RTCA document DO-254 and EUROCAE document ED-80 for simple electronic hardware, such as simple custom micro-coded components and devices, to assist applicants and developers in their demonstration of compliance, and to ensure their safe implementation in airborne systems.

2.0 Background

RTCA document DO-254 (EUROCAE document ED-80), *Design Assurance Guidance for Airborne Electronic Hardware*, was published in April 19, 2000. However, this document has only been recently recognized by some of the certification authorities as an acceptable means of compliance for satisfying the relevant regulations (typically, xx.1301, xx.1309 and 33.28, where xx refers to CS/FAR/JAR 23, 25, 27 or 29)), when custom micro-coded components and devices (such as Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs)) are used in airborne systems. Since the date of publication, the aviation community has gained experience using DO-254/ED-80. However, many applicants and developers are uncertain on several points regarding the document's guidance for simple electronic hardware such as simple custom micro-coded devices. Some feel the definition and the guidance for simple hardware components as stated in DO-254/ED-80, are ambiguous and may be interpreted differently among applicants and developers as well as the certification authorities. Currently, there does not exist any harmonized policy or guidance from the certification authorities to specifically address the safety and airworthiness requirements for simple electronic hardware; however, some applicants are proposing to use the guidance in DO-254/ED-80 to obtain certification approval of simple custom micro-coded components and devices that support critical aircraft functions. The certification authorities need to clarify the guidance in DO-254/ED-80 for simple custom micro-coded components and devices to ensure their safe implementation in airborne systems and to ensure consistent application of the guidance.

3.0 References

a. RTCA/DO-254 (EUROCAE ED-80), *Design Assurance Guidance For Airborne Electronic Hardware*;

***NOTE:* This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

b. FAA AC 20-152, *RTCA, Inc., Document RTCA/DO-254, Design Assurance Guidance For Airborne Electronic Hardware;*
c. CAST 27, Clarifications on the use of RTCA Document DO-254 and EUROCAE Document ED-80, *Design Assurance Guidance for Airborne Electronic Hardware*;
d. SAE ARP 4754, *Certification Considerations for Highly-Integrated or Complex Aircraft Systems.*

4.0 Custom Micro-Coded Components

For this paper, the following terminology applies:

> Programmable Logic Device (PLD) - A component that is purchased as an electronic part 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.
>
> Application Specific Integrated Circuit (ASIC) - Integrated Circuits which are developed to implement a function, including, but not limited to: gate arrays, standard cells and full custom components encompassing linear, digital and mixed mode technologies.
>
> Custom micro-coded components – Programmable Logic Devices (PLDs) and Application Specific Integrated Circuits (ASICs).

Custom micro-coded components are often used to support safety critical airborne functions, and failure of these components could lead to catastrophic or hazardous failure conditions, possibly resulting in loss of aircraft and life, or significant damage and injury. The development and design of these components are often as complex as software controlled microprocessor-based systems.

DO-254/ED-80 defines two categories of hardware items.

- Complex electronic hardware.
- Simple electronic hardware.

FAA Advisory Circular, AC 20-152 [Ref. b] and CAST Paper 27 [Ref. c] provide guidance for satisfying the safety requirements and airworthiness regulations when using DO-254/ED-80 for complex custom micro-coded components such as PLDs and ASICs. The clarifications provided in this paper apply when using DO-

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

254/ED-80 for simple electronic hardware or, more specifically, simple custom micro-coded components.

5.0 DO-254/ED-80 Definition of a Simple Electronic Hardware (SEH) Item

Appendix C of DO-254/ED-80 defines a simple hardware item as the following:

> *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.*

Based on the definition above for a simple hardware item, there is confusion by many applicants on what is meant by the words "comprehensive combination of deterministic tests and analyses" and, more specifically, the word "comprehensive" when applied to a simple hardware item.

Also, Appendix C of DO-254/ED-80 defines a complex hardware item as the following:

> *Complex Hardware Item – All items that are not simple are considered to be complex. See definition of Simple Hardware Item.*

The definition above for a complex hardware item mainly relies on the definition of a simple hardware item, which is not clear to some applicants. However, if one was to infer a definition of Complex Hardware Item using the definition of Simple Hardware Item, the definition might be: "*A hardware item is considered complex if a comprehensive combination of deterministic tests and analyses* ***cannot*** *ensure correct functional performance under all foreseeable operating conditions with no anomalous behavior*". In other words, the item is so complex that it is impossible or impractical to completely test and analyze it, so one must rely on design assurance of the item (a rigorous, structured design process) in addition to some appropriate degree of "comprehensiveness" of tests and analyses.

Additionally, Section 1.6, Complexity Considerations, of DO-254/ED-80 states the following:

> *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.*

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

It is important to note that the description above from Section 1.6 differs from the definition of a simple hardware item as stated in Appendix C. Specifically, the description above includes the additional words, "*appropriate to the design assurance leve*l". Often the applicant fails to notice this difference, which implies that the verification coverage for a Level D simple hardware item may not be adequate or appropriate for a Level A simple hardware item. Moreover, Section 1.6 does not provide any additional guidance on determining which specific deterministic tests and analyses may be appropriate for a Level A simple hardware item as opposed to a Level D hardware item. On the other hand, this varying rigor of verification coverage based on the assigned design assurance level of a simple hardware item is consistent with the guidance in DO-254/ED-80 for complex hardware items of different design assurance levels, specifically the guidance in Appendix A, Modulation of Hardware Life Cycle Data Based on Hardware Design Assurance Level, and Appendix B, Design Assurance Considerations for Levels A and B Functions. This is similar to the guidance for software design assurance where, for example, there is different structural coverage criteria for Software Levels A, B, C and D based on the potential failure conditions to which the software could cause or contribute as determined by a system safety assessment.

6.0 Two Design Assurance Approaches within DO-254/ED-80

As shown in Section 5.0 of this paper, the distinction between simple and complex hardware items is not rigorously defined nor described in DO-254/ED-80. Nevertheless, the guidance within DO-254/ED-80 results in two different methods of showing compliance: relying on a comprehensive combination of deterministic testing and analysis for simple hardware items, or relying on a disciplined hardware design assurance process (i.e., satisfying the objectives of Section 2 through Section 9 of DO-254/ED-80) for complex hardware items.

In many instances, testing of custom micro-coded components or programmed electronic hardware devices cannot demonstrate that the device is free from design defects and errors, because the device is too complex to comprehensively and deterministically test and analyze in a feasible and practical manner. Therefore, DO-254/ED-80 specifies a disciplined, structured design assurance approach for complex electronic devices. On the other hand, simple electronic devices are such that a comprehensive combination of deterministic testing and analysis appropriate to the design assurance level can demonstrate that the device performs its intended function and contains neither design errors nor any anomalous behavior. In other words, if a structured design approach is not utilized, then the simple devices must be evaluated by a comprehensive

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

combination of testing and analysis to ensure correct functionality without design errors or unexpected behavior.

There is an inherent risk for certification programs hidden in the two approaches for design assurance offered by DO-254/ED-80: the decision whether to treat a device as simple or as complex must be made early in the program. Early in the program, a device could be classified by the applicant as simple although it may not be certain that it is practically feasible to demonstrate correct functional performance and absence of anomalous behavior through testing and analysis alone. If later in the program, the applicant finds that it is not possible or practical (i.e., within the constraints of the program schedule) to demonstrate correct functional performance under all foreseeable operating conditions with no anomalous behavior through a comprehensive combination of deterministic tests and analyses, the device must be reclassified as complex. However, this reclassification of the device will most likely require that the development activities for the device will have to be repeated to produce the necessary life cycle data to demonstrate that a disciplined and structured design assurance approach required by DO-254/ED-80 for complex electronic devices has been followed. Therefore, should an applicant classify a device as simple, the applicant should demonstrate the feasibility of the required verification coverage for the device in the hardware plans and obtain agreement early with the cognizant certification authority on the approach to avoid risks to the certification program.

7.0 Clarifications to DO-254/ED-80 for Simple Electronic Hardware

Although it is generally accepted that most modern custom micro-coded devices, such as PLDs, FPGAs and ASICs, are complex electronic devices, especially those FPGAs that have thousands of configurable logic cells used to support multiple functions at the airborne system level, some applicants have proposed their FPGAs as “simple” electronic devices. The following clarification on simple electronic devices is based on information that has been provided to applicants from the certification authorities by way of past Issue Papers, Certification Review Items, and Certification Memos. This guidance is necessary because DO-254/ED-80 provides limited guidance for demonstrating that simple electronic devices comply with the applicable certification requirements (i.e., intended function and freedom from anomalous behavior). Also, the basis for this clarification is to provide guidance for certification authorities and applicants for an appropriate means of demonstrating compliance for simple electronic hardware, which is also not addressed in existing policy and guidance for airborne electronic hardware, including FAA Advisory Circular, AC 20-152 [Ref. b]. This will allow a more standardized approach and “level playing field” for certification

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

authorities, applicants and system developers when implementing simple electronic hardware in airborne systems.

Note: The terminology "comprehensive combination of deterministic testing and analysis" is used in the paragraphs that follow. The definition of the term "comprehensive" is dependent upon the assessed Design Assurance Level (DAL) for the simple device. Due to the increasing criticality of correct operation of the device as the hazard classification of the device increases, the rigor and thoroughness of testing a simple device is necessarily greater for a DAL A or B device than for a DAL C or D device. This is analogous to the increasingly rigorous design assurances processes documented in RTCA DO-178B (EUROCAE ED-12B) for airborne software and DO-254/ED-80 for complex custom micro-coded components.

The following items should be addressed for SEH:

a. Level A and B: The SEH associated with functions whose failure or malfunction could cause a catastrophic failure condition or a hazardous/severe-major failure condition (i.e., hardware design assurance Levels A or B), as determined by the system safety assessment process, require the following:

(1) A comprehensive combination of deterministic tests and analysis that demonstrates correct operation under all possible combinations, permutations, and concurrence of conditions across all primary inputs, internal elements, nodes, registers, latches, logic components, gates, etc. within the device with no anomalous behavior. If the inputs of some specific gates, nodes, etc., cannot be sufficiently controlled using external device inputs to create all required transitions, then the testing should be augmented by additional techniques to accomplish acceptable device stimulation. If the outputs of some specific gates, nodes, etc., cannot be sufficiently observed using external device outputs to detect correct operation, then the testing should be augmented by additional observation techniques, such as design assertions (e.g. Open Verification Library, System Verilog assertions, or Property Specification Language, etc.), scan chain testing methodologies, or other techniques, to accomplish acceptable coverage. See Appendix A, Frequently Asked Questions, Question #2 for more information.

(2) A test coverage analysis to ensure that the testing and analyses satisfy the specified criteria and are complete. The test coverage analysis should confirm that all logical gate/nodes within the device, as well as the interconnections between these gates/nodes, have indeed been exercised in a manner which demonstrates proper operation of the elements within the

***NOTE:* This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

device. For example, an "OR" gate should be tested in a manner which demonstrates it truly operates as an "OR" gate. Additionally, all possible states of a sequential state machines and if applicable, all combinations of possible states of multiple state machines should be shown to have been tested. If concurrency is present in the device, then all possible concurrency conditions should be shown to have been tested. Concurrency will be present any time a device has multiple independent data streams that interact together in some way through shared resources, arbiters, multiple interacting state machines, etc.

(3) Timing analysis should cover best-case and worst-case timing conditions, potential clock drift, and other timing issues that may prevent correct operation of the device. Also, adverse environmental conditions such as temperature should be considered in this timing analysis.

b. Level C: The SEH associated with functions whose failure or malfunction could cause a major failure condition as determined by the system safety assessment (design assurance level C hardware) should undergo a comprehensive combination of deterministic testing and analysis that demonstrates correct operation under all possible combinations and permutations of conditions of the inputs at the pins of the device (i.e., those inputs available external to the packaging of the device). All possible states of any sequential state machines should also be tested.

c. Level D: The SEH associated with functions whose failure or malfunction could cause a minor failure condition for the airplane as determined by the system safety assessment (DAL D hardware) may be tested at the equipment level to demonstrate the device performs as required. That is, testing of the card, module, or Line Replaceable Unit (LRU) in which the SEH is installed may be used to show that the SEH satisfies the device level requirements with the same test procedures used to verify correct operation of the card, module, or LRU. This approach should be documented in the system certification plan [Ref. d., Section 4.4.1] or in the Plan for Hardware Aspects of Certification [Ref. a., Section 10.1.1].

d. Documentation: Section 1.6 of DO-254/ED-80 states the following:

> *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.*

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

To clarify the requirements for documentation of simple custom micro-coded devices, the following documentation should be submitted to the cognizant certification authority for all hardware design assurance levels (A-D):

1. Plan for Hardware Aspects of Certification (PHAC) [Ref. a., Section 10.1.1]
2. Hardware Verification Plan [Ref. a., Section 10.1.4]
3. Hardware Configuration Index [Ref. c., Section 10b]
4. Hardware Accomplishment Summary [Ref. a., Section 10.9]

The above documentation can be combined with the other submitted documentation for complex custom micro-coded devices when using DO-254/ED-80. For example, a single PHAC for both simple and complex devices may be submitted to the certification authority. Also, test procedures, test cases, test results and test coverage analyses for simple devices should be documented and retained as verification life cycle data and are subject to certification authority review upon request.

8.0 Certification Authorities Software Team (CAST) Position

The applicant and certification authority should ensure that the clarifications to RTCA/DO-254 (EUROCAE/ED-80) provided in this paper are addressed when DO-254/ED-80 is the proposed approach to obtain approval of electronic hardware (including simple electronic hardware), such as simple custom micro-coded devices, used in airborne systems and equipment. Alternate methods or processes in lieu of DO-254/ED-80 should be approved by the certification authority prior to their implementation.

APPENDIX A: Frequently Asked Questions (FAQs)

1. What is concurrency and what problems does it introduce when attempting to comprehensively test a SEH device?

Concurrency occurs when two or more execution flows are able to run simultaneously and which may permit the sharing of common resources between those overlapped computations. Race conditions involving these multiple execution flows can result in unpredictable system behavior.

Concurrency can be identified by a variety of “signatures”, including (but not limited to):

a. Multiple independent data paths through the device
b. The presence of an arbiter

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

c. The presence of multiple state machines, especially interacting state machines
d. The presence of an asynchronous interrupt

Given two designs of equal numbers of inputs and state elements, one with concurrency and one without, it becomes likely that the design with concurrency will contain a greater number of design errors over the non-concurrency design. Also, the verification of a design with concurrency will be more difficult. Given this, the number of test cases required to exhaustively verify a design with concurrency will be much greater. This reduces the number of inputs and/or internal states a design can have in order to reasonably assure exhaustive verification.

2. What does "comprehensive verification" of a DAL A or B SEH device entail?

In order to comprehensively verify the functionality of a DAL A or a DAL B SEH device, the following should be accomplished:

1. Guarantee generation and application of comprehensive stimulation of the device.
    a. Provide all possible stimuli to every element in the design, where an "element" is one of the following:
        i. External input
        ii. Internal register
        iii. Internal latch
        iv. Logic element (e.g. AND gate, OR gate, etc)
    b. Guarantee exhaustive stimulus of all possible device inputs has been generated and applied in combination with all possible internal states within state elements (e.g., registers, latches), as well as across all possible concurrency events. That is, skew all input combinations across all possible times in order to guarantee operation with respect to all possible concurrency. This includes skewing of all possible input conditions against all other possible input combinations, in combination with all state elements holding all possible state values, in combination with all state elements transitioning between all possible state transitions.

2. Verify the correct performance of each "element" in the presence of all possible stimuli. This includes:
    a. Observe correct results of each design element under all stimulus conditions.

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

b. Observations must be made using a primary device output, or using other acceptable internal observation methods, such as design assertions (such as Open Verification Library, System Verilog Assertions, or Property Specification Language), scan chain testing methodologies, or other techniques.

> **Note:** For assertion-based verification, coverage metrics, such as minimum sequential distance and assertion density, should be considered to determine whether the design needs more assertions. Assertion density measures the number of assertions of each type in each module. Minimum sequential distance measures the minimum number of levels of sequential logic from a given register to any assertion. For example, an adder consists of a variety of AND and OR gates. One can use design assertions to observe the inputs and outputs of the adder to determine correct operation instead of observing each internal node within the adder.

3. Timing analysis should cover best-case and worst-case timing conditions, potential clock drift, and other timing issues that may prevent correct operation of the device.

3. When is comprehensive verification impractical?

There are significant challenges to proving comprehensive verification have been accomplished. In order to make the assertion that a device can be comprehensively verified, one must be reasonably sure comprehensive verification is theoretically possible for a given device.

Comprehensive verification becomes virtually impossible under several conditions:

1. Concurrency exists within the device. The presence of concurrency introduces the likelihood of race conditions which may not be uncovered with a minimum set of test cases. Additionally, the existence of multiple state machines causes the number of possible overall states of the device to increase exponentially.

2. Required stimulus exceeds the capabilities of available testing methods. If the combination of inputs and internal states creates a need for excessive stimulation, (e.g., greater than six months of continuous runtime), it

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

should be assumed that such a device cannot be comprehensively tested (see FAQ 4. below for example).

4. How many test cases are required for non-concurrent devices?

The minimum number of test cases for a non-concurrent device (applying all possible input combinations in conjunction with all possible internal states) equates to:

$$\text{Min number of test cases} = 2^{\#\text{ inputs}} \times 2^{\#\text{ state elements}} = 2^{(\#\text{inputs} + \#\text{ state elements})}$$

For example, a device with 5 inputs and 3 registers would equate to 256 test cases.

$$2^5 \times 2^3 = 32 \times 8 = 256 \text{ required test cases.}$$

Note: This calculation is not exact, as most designs will likely need several sequential test cases to change the internal state prior to applying the suite of input vectors. However, this will suffice to offer an upper bound of when exhaustive testing is tractable.

Additionally, a device with a total of 44 inputs and state elements and a fast test environment able to input one new vector per microsecond would require 6.79 months of continuous testing (e.g., 1.76×10^{13} test cases, or 2^{44} test vectors).

NOTE: **This position paper has been coordinated among representatives of certification authorities from North and South America, and Europe. However, <u>it does not constitute official policy or guidance from any of the authorities</u>. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**