U.S. Department
of Transportation
Federal Aviation
Administration

Advisory Circular

Subject: Use of Multi-Core Processors

Date: 1/8/24

AC No: 20-193

Initiated By: AIR-622

1 PURPOSE.

1.1 This advisory circular (AC) describes an acceptable means of compliance with the applicable airworthiness regulations for multi-core processors (MCP) contained in airborne systems and equipment used in type certification or technical standard order (TSO) authorization. The contents of the document do not have the force and effect of law and are not meant to bind the public in any way. The document is intended only to provide information to the public regarding existing requirements under the law or agency policies.

1.2 This AC provides objectives for compliance demonstration with the applicable airworthiness regulations for airborne systems and equipment that contain MCPs, according to the applicability in paragraph 2 of this document.

2 APPLICABILITY.

2.1 This AC may be used by applicants and design approval holders of airborne systems and equipment that contain MCPs to be installed on type certificated aircraft, engines, and propellers, or to be used in TSO articles. This also includes suppliers that are involved in the software or airborne electronic hardware (AEH) life cycle processes described in this AC. The applicant or design approval holder is responsible for oversight of all its suppliers.

2.2 This AC applies to systems and equipment that contain MCPs with two or more activated cores for which the item development assurance level (IDAL) of at least one of the software applications hosted by the MCP, or of the hardware item containing the MCP, is A, B, or C. The deactivation of cores is handled through the applicable AEH guidance.

2.3 The AC does not apply when the IDALs are all level D or E.

2.4 If an applicant modifies the use of the MCP (such as by activating one or more additional cores or adding software of IDAL A, B, or C), then the applicant should reassess the applicability of this AC.

2.5 Paragraph 5.7 of this document describes the objectives that apply according to the assigned IDAL (A, B, or C) of the hosted software or of the hardware item containing the MCP.

2.6 Aspects not covered by this AC.

2.6.1 The features described in paragraphs 2.6.1 and 2.6.2 are not covered by the guidance in this AC. The other aspects of this AC may still be applicable if an applicant uses these features. Additional project specific guidance may be applied if necessary.

2.6.2 Any applicant who proposes to use these features should describe how they are used so that the behavior of the MCP is not altered, and determinism is still guaranteed.

2.6.3 In their planning activities, the applicant should present the methods employed to cover these aspects and satisfy the objectives of this AC or show compliance with the applicable airworthiness regulations, if they propose an alternative to this AC or part of it.

2.6.4 Dynamic allocation of software applications.

2.6.4.1 This AC does not cover MCP platforms on which software applications or tasks can be dynamically re-allocated to a different core(s) by the operating system, a software hypervisor, or by other means.

Note: This AC assumes that software applications are statically allocated to cores during MCP startup but not during the subsequent operation of the software.

2.6.4.2 However, justification for using dynamic allocation features within the scope of this AC may rely on robust and proven limitations that lead to deterministic behavior, such as: Restricted usage permitting the applicant to claim equivalence to the conditions expressed in this AC (for example multi-static allocation, i.e., selection of a prequalified configuration, instead of pure dynamic allocation).

2.6.5 Simultaneous multithreading support within processors.

This AC does not cover simultaneous multithreading, as industry and the certification authorities' knowledge and experience of such features are currently insufficient to provide AC guidance for their certification. This issue is not specific to MCPs.

2.7 Exceptions.

An MCP may contain multiple cores of different types, which may interact in different ways; some of the interactions do not produce interference. Therefore, the objectives of this AC do not apply to the interactions between two or more activated cores of an MCP in the following cases:

- The activated cores are set up in lock-step mode; or
- The activated cores are only linked by the conventional databuses typically used in avionic systems, and not by any of the following: shared memory, shared cache, or a “coherency fabric” or “module interconnect”. This category includes the case where the cores only act as co-processors or graphics processors, each under the control of another core that executes software.

The objectives of this AC apply to the interactions between all the other activated cores of an MCP.

3 BACKGROUND.

3.1 MCPs can execute several software applications at the same time by hosting them on different cores. Therefore, several software applications and/or hardware functions may attempt to access the same MCP shared resources (such as memory, cache, “coherency fabric”, or “module interconnect”, or external interfaces) at the same time, causing contention for those resources.

3.2 Most MCPs have internal features to handle and arbitrate the concurrent demands for MCP resources, which may cause delays in access to the resources. These delays are a form of time interference between the software applications or tasks, which can cause the software applications to take much longer to execute than when executing on their own.

3.3 The execution of software applications may be different on MCPs than it is on single-core processors (due to parallelism and other MCP mechanisms, or software components such as operating systems or hypervisors). This may result in new or different data or control coupling paths, and functional interference between the software applications or tasks.

3.4 Interference between the software applications or tasks executing on an MCP could cause safety-critical software applications to behave in a non-deterministic or unsafe manner or could prevent them from having sufficient time to complete the execution of their safety-critical functionality.

4 DEFINITIONS.

- Applicable airborne electronic hardware (AEH) guidance: AC 20-152*Development Assurance for Airborne Electronic Hardware*, and any project-specific guidance.
- Applicable software guidance: AC 20-115, *Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( )*, and any project-specific guidance.
- Asymmetric multi-processing (AMP): an MCP software architecture in which each individual functional task is permanently allocated to a specific core and each core has its own operating system (however, the operating systems may be multiple copies of the same operating system or be different from core to core).
- Bound multi-processing (BMP): an MCP software architecture that extends the symmetric multi-processing architecture by allowing tasks to be bound to specific cores while using a common operating system across all cores.
- Determinism/deterministic: the ability to produce a predictable outcome generally based on the preceding operations and data. The outcome occurs in a specific period of time with repeatability. (Definition taken from ED-124/DO-297).
- Hardware component: any part of the hardware which may independently access MCP shared resources.
- Integrated Modular Avionics (IMA) platform: an integrated modular avionics MCP platform that provides both robust resource partitioning and robust time partitioning (as defined in this document).
- Intended final configuration: the configuration of the software and hardware in which the set of MCP resources has been defined by implementing the configuration settings and all software components have been installed on the target MCP.
- Interference channel: a platform property that may cause interference between software applications or tasks.
- Item: a hardware or software element having bounded and well-defined interfaces (Definition taken from ED-79A/ARP4754A).
- Item development assurance level (IDAL): the level of rigor for development assurance tasks performed on item(s) is the appropriate software level in ED-12C/DO-178C and design assurance level in ED-80/DO-254 objectives that need to be satisfied for an item. (Definition taken from ED-79A/ARP4754A).
- MCP platform: consists of the MCP itself and, in many cases, the platform software, such as an operating system and/or software hypervisor, which provides the interface between the software applications and the MCP.

- MCP platform with robust partitioning: an MCP platform that complies with the objectives of this document and provides robust resource partitioning and robust time partitioning as defined in this document, not only between software applications hosted on the same core, but also between software applications hosted on different cores of an MCP or between software applications that have tasks hosted on several cores.
- Multi-core processor (MCP): an AEH device that contains two or more processing cores. A core in an MCP is defined as a device that executes software. This includes virtual cores (e.g., in a simultaneous multithreading microarchitecture). An MCP is typically implemented in a device that may also include resources such as memory or peripheral controllers, internal memory, peripherals, and internal interconnects.
- Robust partitioning: both robust resource partitioning and robust time partitioning.
- Robust resource partitioning (adapted from ED-94C/DO-248C and ED-124/DO-297): robust resource partitioning is achieved when:
  - Software partitions cannot contaminate the storage areas for the code, I/O, or data of other partitions.
  - Software partitions cannot consume more than their allocations of shared resources; and
  - Failures of hardware unique to a software partition cannot cause adverse effects on other software partitions.

  Note: Software that provides partitioning should have at least the same IDAL as the highest IDAL of the software that it partitions.
- Robust time partitioning (on an MCP): this is achieved when, as a result of mitigating the time interference between partitions hosted on different cores, no software partition consumes more than its allocation of execution time on the core(s) on which it executes, irrespective of whether partitions are executing on none of the other active cores or on all of the other active cores.
- Safety net: the employment of mitigations and/or protections at the appropriate level of aircraft and system design to satisfy the safety objectives. The safety net methodology may be applied when it is assumed that part of a system will misbehave. The safety net is independent of the source of misbehavior. The safety net can include passive monitoring functions, active fault avoidance functions, and control functions for effective recovery of system operations from anomalous events.
- Software application: generally, designates the software part of a function installed on an MCP.
- Software component: any part of the software which may access MCP shared resources. It may designate either a software application or an operating system or a hypervisor.

- Symmetric multi-processing (SMP): an MCP software architecture in which a single operating system controls the execution of the software on multiple cores and may dynamically allocate tasks to cores at run-time.
- Task: the smallest unit of software execution that can be managed independently by a scheduler. For the purpose of this document, this term encompasses “threads” or “processes” (in the sense of ARINC 653). For simplification in this AC, when addressing interference, a task also represents any part of an application or any part of a software component that executes on one core.

5 MULTI-CORE PROCESSOR GUIDANCE.

5.1 There are important issues involved in each stage in the typical life cycle of an MCP project. Applicants are expected to meet certain objectives for each stage, except for any objective, or part of an objective, that the applicant justifies as not being applicable to the MCP in their system or equipment, (e.g., if the MCP mechanism addressed does not exist on the selected MCP). The applicant should state, in the appropriate deliverable document, which aspects do not apply and explain why they do not apply.

Note: Some of the objectives have notes provided after them. These notes should be part of the objectives, as they provide additional information that is relevant to the objectives. Objectives and their included notes are formatted in italics to differentiate them from the rest of the text.

5.2 Planning objectives.

5.2.1 The additional planning objectives below clarify the information to be included in the applicable plans to achieve planning data standardization for projects with MCPs.

Objective MCP_Planning_1:

The applicant’s plans or other deliverable documents:

1. Identify the specific MCP processor, including the unique identifier from the manufacturer.

2. Identify the number of active cores.

3. Identify the MCP software architecture to be used and all the software components that will be hosted on the MCP.

4. Identify any dynamic features provided in software hosted on the MCP that will be activated and provide a high-level description of how they will be used.

5. Identify whether the MCP will be used to host software applications from more than one system, and whether it will be used in an integrated modular avionics (IMA) platform.

6. Identify whether the MCP platform will provide robust resource partitioning and / or robust time partitioning as defined in this document.

7. Identify the methods and tools to be used to develop and verify all the individual software components hosted on the MCP so as to meet the objectives of this document and the applicable software guidance, including any methods or tools needed due to the use of an MCP or the selected MCP architecture.

NOTES:

a) The MCP software architecture includes asymmetric multi-processing (AMP), symmetric multi-processing (SMP) or any other architecture used by the applicant.

b) The software components identified should include any operating systems, hypervisors, software applications, and all functions that are provided in software. In the case of an MCP used in an IMA platform, the software components that are identified do not have to include the hosted software applications.

c) The dynamic features provided in software should include such aspects as the dynamic allocation of software applications or tasks to cores and any other software dynamic features that can affect the execution of the software while it is executing.

5.2.2 Multiple software applications and/or hardware functions may use resources of the MCP and may cause contention for resources and interference between software applications or tasks. Even if there is no explicit data or control flow between software applications or tasks running concurrently on different cores, MCP resources (e.g., cache or interconnects) may be shared. Therefore, coupling may exist on the platform level which can cause interference between the software applications or tasks and cause increases in the worst-case execution times (WCETs) of the software applications. In addition, there could be interaction between software and hardware functions that would need to be considered (e.g., cases where there are multiple masters).

Objective MCP_Planning_2:

The applicant's plans or other deliverable documents:

1. *Provide a high-level description of how MCP shared resources will be used and how the applicant intends to allocate and verify the use of shared resources (see Note a) to avoid or mitigate the effects of contention for MCP resources and to prevent the resource capabilities of the MCP from being exceeded by the demands from the software applications and/or the hardware components of the MCP.*
2. *Identify the MCP hardware resources to be used to support the objectives in this AC.*
3. *Identify any hardware dynamic features of the MCP that will be active and provide a high-level description of how they will be used.*
4. *Identify the aspects of the use of the MCP that may require a safety net or other mechanisms to detect and handle failures in the MCP.*

NOTES:

a) The description of the use of shared resources should include any use of shared cache (taking into account the time interference it may cause due to cache misses or other effects) or shared memory (taking into account the time interference and the data and control flow effects it may cause such as lockouts, race conditions, data starvation, deadlocks, live-locks, or excessive data latency). The description of shared resources should also include any use of shared interconnect and take into account the time interference due to arbitration for access to the shared interconnect.

b) Hardware dynamic features of the MCP include any features that can alter the behavior of the MCP or the hosted software during execution, for example, energy-saving features (clock enable/ gating, frequency adaptations, deactivating one or more cores, or dynamic control of peripheral access).

5.3 Setting of MCP resources.

In the context of MCPs, some of the configuration settings are especially relevant to the MCP hardware and software architectures, such as:

- which cores are activated,
- the execution frequencies of the cores,
- the priorities and allocation of shared interconnect,
- which of the peripheral devices of the MCP are activated,
- whether shared memory or shared cache is used and how each is allocated, and
- Whether dynamic features that are built into some MCPs are allowed to alter the frequency of execution of the cores or to deactivate one or more cores in order to save energy. (This might not be desirable for cores hosting safety-critical software applications.)

Objective MCP_Resource_Usage_1:

The applicant has determined and documented the MCP configuration settings that will enable the hardware and the software hosted on the MCP to satisfy the functional, performance, and timing requirements of the system.

Objective MCP_Resource_Usage_2:

Reserved. Covered by AC 20-152A objective COTS-8.

5.4 Interference channels and resource usage.

5.4.1 The software applications or tasks that execute on different cores of a multi-core processor share MCP resources, so even if there is no explicit data or control flow between these software applications or tasks, coupling exists on the platform level, which can cause interference between them.

5.4.2 There may be software or hardware channels through which the MCP cores or the software hosted on those cores could interfere with each other, in addition to those channels specifically mentioned in this AC. For instance, many MCPs include an “interconnect” or “coherency fabric”, through which the demands for MCP resources, e.g., from the software applications hosted on the MCP, are channeled and the demands are arbitrated. This arbitration can cause interference effects such as jitter on data arrival times, data consistency issues, or it can change the order in which transactions requested by the software applications are executed.

5.4.3 Non-deterministic behavior of the hosted software applications may occur due to such interference.

5.4.4 Moreover, the complexity of the MCP, executing tasks in parallel, and the interference could lead to the demands for resources exceeding the available resources. For instance, if the demands for interconnect transactions are very high in MCPs with a very high level of external databus traffic, the interconnect can become overloaded, which can affect transactions on some MCPs.

MCP_Resource_Usage_3:

The applicant has identified the interference channels that could permit interference to affect the software applications hosted on the MCP cores and has verified the applicant’s chosen means of mitigation of the interference.

NOTES:

a) This objective includes the identification of any interference caused by the use of shared memory, shared cache, an interconnect, or the use of any other shared resources, including shared peripherals, and the verification of the means of mitigation chosen by the applicant.

b) If the applicant identifies interference channels that cannot affect the software applications in the intended final configuration, then those interference channels do not need to be mitigated and no verification of mitigation is needed.

c) The applicant should handle any interference channel discovered at any time during the project in the same manner as in this objective and these explanatory notes.

d) If the highest IDAL of the MCP hardware and of all the software applications hosted on the MCP is C and the hosted software applications are not required by the safety analysis to be robustly partitioned, then the applicant has the option to not conduct an interference analysis and therefore to not meet this objective. However, applicants should note that opting to not meet this objective affects the way they are permitted to conduct their software verification. (See objective MCP_Software_1 and Note c) of that objective.)

MCP_Resource_Usage_4:

The applicant has identified the available resources of the MCP and of its interconnect in the intended final configuration, has allocated the resources of the MCP to the software applications hosted on the MCP, and has verified that the demands for the resources of the MCP and of the interconnect do not exceed the available resources when all the hosted software is executing on the target processor.

NOTE: *The use of worst-case scenarios is implicit in this objective.*

5.5 **Software verification.**

5.5.1 The software verification processes in the applicable software guidance need to be adapted for use on an MCP to demonstrate that the hosted software applications function correctly and have sufficient time to execute in the presence of the interference that occurs when all the hosted software is executing on an MCP.

5.5.2 With an MCP, there may be data and control flows between software components or tasks hosted on different cores of the MCP. Therefore, the data and control coupling analysis performed on the software hosted on each separate core (as requested by the applicable software guidance) may not reveal the improper software behavior associated with features such as hardware runtime optimizations and memory models on MCPs.

5.5.3 The WCET of a software component or task may increase significantly when other software components or tasks are executing in parallel on the other cores of an MCP. This could cause some software applications to have insufficient time to complete the execution of their safety-critical functionality.

5.5.4 Interference and interactions between software applications or tasks occur via the proprietary internal mechanisms of an MCP. Any simulation of those mechanisms is therefore less likely to be representative in terms of functionality or execution time than testing conducted on the target MCP in the intended final configuration, and thus is less likely to detect errors.

5.5.5 To adapt the software verification guidance for different types of MCP platforms, the two following categories of MCP platforms are considered:

- MCP platforms with robust partitioning, and
- All other MCP platforms.

MCP_Software_1:

The applicant has verified that all the software components hosted by the MCP meet the objectives of the applicable software guidance. In particular, the applicant has verified that all the hosted software components function correctly and have sufficient time to complete their execution when all the hosted software and hardware of the MCP is executing in the intended final configuration.

The way in which the applicant should satisfy this objective depends on the type of the MCP platform:

- ***MCP platforms with robust partitioning:***

  Applicants who have verified that their MCP platform provides both robust resource partitioning and robust time partitioning (as defined in this document) may verify software applications separately on the MCP and determine their WCETs separately.

- ***All other MCP platforms:***

  Applicants may verify separately on the MCP any software component or set of requirements for which the interference identified in the interference analysis is mitigated or is precluded by design. Software components or sets of software requirements for which interference is not avoided or mitigated should be tested on the target MCP with all software components executing in the intended final configuration, including robustness testing of the interfaces of the MCP.

  The WCET of a software component may be determined separately on the MCP if the applicant shows that time interference is mitigated for that software component; otherwise, the WCET should be determined by analysis and confirmed by test on the target MCP with all the software components executing in the intended final configuration.

NOTES:

a) All the interfaces between the hosted software and the hardware of the MCP should be included in this testing.

b) The robustness testing mentioned above is intended to cover the specific aspects of an MCP that are not specifically covered by the standard verification activities described in the applicable software guidance.

c) If the highest IDAL of the MCP hardware and of all the software applications hosted on the MCP is C and the hosted software applications are not required by the safety analysis to be robustly partitioned, then the applicant has the option to not conduct an interference analysis and therefore to not meet objective MCP_Resource_Usage_3. In such a case where no interference analysis has been performed, the hosted software components should be verified according to this objective as components for which interference is not avoided or mitigated and for which separate verification is therefore not permitted.

d) To "verify separately" and "determine the WCET separately" mean to conduct these activities without all the software executing at the same time on other cores of the MCP.

e) Interference may occur between tasks of a single component when the tasks execute on different cores.

MCP_Software_2:

The applicant has verified that the data and control coupling between all the individual software components hosted on the same core or on different cores of the MCP has been exercised during software requirement-based testing, including exercising any interfaces between the software components via shared memory and any mechanisms to control the access to shared memory, and that the data and control coupling is correct.

NOTES:

a) When this objective cannot be completely met during the software verification, applicants may propose to use system level testing to exercise the data and control coupling between software components hosted on different cores.
b) Interference may occur between tasks of a single component when the tasks execute on different cores (see definition of task).

5.6 Error detection and handling, and safety nets.

5.6.1 In addition to the types of errors and failures normally detected and handled in a system that incorporates a single-core processor, additional types of errors and failures may need to be detected and handled in an MCP environment due to problems caused by the features of MCPs and due to the additional complexity of executing several software applications or tasks in parallel in real time.

5.6.2 Features of an MCP may therefore contain unintended functionality that may cause errors and produce unexpected behavior. Applicants may therefore wish to consider the use of a “safety net” independent from the MCP to detect and handle failures within the MCP and to contain any such failures within the equipment in which the MCP is installed.

MCP_Error_Handling_1:

The applicant has identified the effects of failures that may occur within the MCP and has designed, implemented, and verified means commensurate with the safety objectives, by which to detect and handle those failures in a fail-safe manner that contains the effects of any failures within the equipment in which the MCP is installed. These means may include a “safety net” independent from the MCP.

5.7 Data to complement the accomplishment summaries.

The applicant is expected to describe how the objectives of this AC were satisfied.

MCP_Accomplishment_Summary_1:

In addition to providing the information requested by the applicable software and AEH guidance, the applicant has provided documentation that summarizes how they have met each of the objectives of this document.

5.8 Applicability of the MCP objectives according to their IDALs.

5.8.1 The column "IDAL A or B" shows the objectives applicable when the highest IDAL of any of the software applications hosted by the MCP or of the MCP hardware device is A or B.

5.8.2 The column "IDAL C" shows the objectives applicable when the highest IDAL of any of the software applications hosted by the MCP or of the MCP hardware device is C.

MCP OBJECTIVES	IDAL A or B	IDAL C
MCP_Planning_1	Yes	Yes
MCP_Planning_2	Yes	Yes
MCP_Resource_Usage_1	Yes	Yes
MCP_Resource_Usage_2	n/a	n/a
MCP_Resource_Usage_3	Yes	Refer to NOTE d
MCP_Resource_Usage_4	Yes	No
MCP_Software_1	Yes	Yes
MCP_Software_2	Yes	Yes
MCP_Error_Handling_1	Yes	No
MCP_Accomplishment_Summary_1	Yes	Yes

6 RELATED READING MATERIAL

6.1 Title 14, Code of Federal Regulations (14 CFR).

This AC provides guidance on development of an acceptable means of compliance to the following regulations, with respect to the development assurance of software and AEH.

- Part 21, *Certification Procedures for Products and Articles.*
- Part 23, *Airworthiness Standards: Normal Category Airplanes.*
- Part 25, *Airworthiness Standards: Transport Category Airplanes*.
- Part 27, *Airworthiness Standards: Normal Category Rotorcraft*.
- Part 29, *Airworthiness Standards: Transport Category Rotorcraft*.
- Part 33, *Airworthiness Standards: Aircraft Engines*.

- Part 35. *Airworthiness Standards: Propellers*.
- Section 21 subpart O, *Technical Standard Order Approvals*.
- Section 23.2500, *Airplane level systems requirements.*
- Section 23.2505, *Function and installation*.
- Section 23.2510, *Equipment, systems, and installations*.
- Section 25.1301, *Function and installation*.
- Section 25.1309, *Equipment, systems, and installations*.
- Section 27.1301, *Function and installation*.
- Section 27.1309, *Equipment, systems, and installations*.
- Section 29.1301, *Function and installation*.
- Section 29.1309, *Equipment, systems, and installations*.
- Section 33.28, *Engine control systems*.
- Section 35.23, *Propeller control systems*.

6.2 **FAA Advisory Circulars.**

- AC 20-115, *Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( ).*
- AC 20-152, *Development Assurance for Airborne Electronic Hardware (AEH).*
- AC 20-170, *Integrated Modular Avionics Development, Verification, Integration and Approval using RTCA DO-297 and Technical Standard Order C153*.
- AC 20-171, *Alternatives to RTCA/DO-178B for Software in Airborne Systems and Equipment*.
- AC 20-174, *Development of Civil Aircraft and Systems*.
- AC 20-189, *Management of Open Problem Reports (OPRs)*.
- AC 21-46, *Technical Standard Order Program*.
- AC 21-50, *Installation of TSOA Articles and LODA Appliances*.
- AC 23.1309-1, *System Safety Analysis and Assessment for Part 23 Airplanes*.
- AC 23.2010-1, *FAA Accepted Means of Compliance Process for 14 CFR Part 23*.
- AC 25.1309-1, *System Design and Analysis*.
- AC 27-1, *Certification of Normal Category Rotorcraft.*
- AC 29-2, *Certification of Transport Category Rotorcraft.*

- AC 33.28-1, *Compliance Criteria for 14 CFR § 33.28, Aircraft Engines, Electrical and Electronic Engine Control Systems*.
- AC 33.28-2, *Guidance Material for 14 CFR § 33.28, Reciprocating Engines, Electrical and Electronic Engine Control Systems*.
- AC 33.28-3, *Guidance Material for 14 CFR § 33.28, Engine Control Systems*.
- AC 35.23-1, *Guidance Material for 14 CFR § 35.23, Propeller Control Systems*

6.3 **EASA Acceptable Means of Compliance.**

- AMC 20-115( ), *Airborne Software Development Assurance Using EUROCAE ED-12 and RTCA DO-178*.
- AMC 20-152( ), *Development Assurance for Airborne Electronic Hardware (AEH).*

6.4 **Industry Documents.**

- SAE International Aerospace Recommended Practice (ARP) 4754A, *Guidelines for Development of Civil Aircraft and Systems*, dated December 21, 2010.
- RTCA DO-178, *Software Considerations in Airborne Systems and Equipment Certification*, dated January 1982 (no longer in print).
- RTCA DO-178A, *Software Considerations in Airborne Systems and Equipment Certification*, dated March 1985 (no longer in print).
- RTCA DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 1, 1992.
- RTCA DO-178C, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 13, 2011.
- RTCA DO-248C, *Supporting Information for DO-178C and DO-278A*, dated December 13, 2011.
- RTCA DO-254, *Design Assurance Guidance for Airborne Electronic Hardware*, dated April 19, 2000.
- RTCA DO-297, *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, dated November 8, 2005.
- RTCA DO-330, *Software Tool Qualification Considerations*, dated December 13, 2011.
- RTCA DO-331, *Model-Based Development and Verification Supplement to DO-178C and DO-278A*, dated December 13, 2011.
- RTCA DO-332, *Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A*, dated December 13, 2011.
- RTCA DO-333, *Formal Methods Supplement to DO-178C and DO-278A*, dated December 13, 2011.

- EUROCAE ED-12, *Software Considerations in Airborne Systems and Equipment Certification*, dated May 1982 (no longer in print).
- EUROCAE ED-12A, *Software Considerations in Airborne Systems and Equipment Certification*, dated October 1985 (no longer in print).
- EUROCAE ED-12B, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 1992.
- EUROCAE ED-12C, *Software Considerations in Airborne Systems and Equipment Certification*, dated January 2012.
- EUROCAE ED-79A, *Guidelines for Development of Civil Aircraft and Systems*, dated December 2010.
- EUROCAE ED-80, *Design Assurance Guidance for Airborne Electronic Hardware*, dated April 2000.
- EUROCAE ED-94C, *Supporting Information for ED-12C and ED-109A*, dated January 2012.
- EUROCAE ED-124, *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, dated June 2007.
- EUROCAE ED-215, *Software Tool Qualification Considerations*, dated January 2012.
- EUROCAE ED-216, *Formal Methods Supplement to ED-12C and ED-109A*, dated January 2012.
- EUROCAE ED-217, *Object-Oriented Technology and Related Techniques Supplement to ED-12C and ED-109A*, dated January 2012.
- EUROCAE ED-218, *Model-Based Development and Verification Supplement to ED-12C and ED-109A*, dated January 2012.

7 WHERE TO FIND THIS AC

You can find this AC in the Dynamic Regulatory System (DRS) at https://drs.faa.gov/browse.

8 SUGGESTIONS FOR IMPROVING THIS AC.

If you have suggestions for improving this AC, you may use the FAA Form 1320-73, *Advisory Circular Feedback Information* at the end of this AC.

DANIEL J. ELGAS

Digitally signed by DANIEL J. ELGAS
Date: 2024.01.08 12:16:17 -05'00'

Daniel J. Elgas
Director, Policy and Standards Division
Aircraft Certification Service

Appendix A. Advisory Circular Feedback Form

Paperwork Reduction Act Burden Statement: A federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with a collection of information subject to the requirements of the Paperwork Reduction Act unless that collection of information displays a currently valid OMB Control Number. The OMB Control Number for this information collection is 2120-0746. Public reporting for this collection of information is estimated to be approximately 20 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering, and maintaining the data needed, completing and reviewing the collection of information.
All responses to this collection of information are voluntary FAA Order 1320.46D Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to: Information Collection Clearance Officer, Barbara Hall, 800 Independence Ave, Washington, D.C. 20590.

If you find an error in this AC, have recommendations for improving it, or have suggestions for new items/subjects to be added, you may let us know by (1) emailing this form to 9-AWA-AVS-AIR-DMO@faa.gov or (2) faxing it to the attention of the LOB/SO (______________________).

Subject: AC 20-193, *Use of Multi-Core Processors* Date: ____________________

Please mark all appropriate line items:

☐ An error (procedural or typographical) has been noted in paragraph ____________ on page _________________________.

☐ Recommend paragraph ____________ on page ____________ be changed as follows:

☐ In a future change to this AC, please cover the following subject:
(Briefly describe what you want added.)

☐ Other comments:

☐ I would like to discuss the above. Please contact me.

Submitted by: ____________________________________Date: __________________________

FAA Form 1320-73 (09/22) SUPERSEDES PREVIOUS EDITIONS