Computer Standards & Interfaces 96 (2026) 104049

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Towards a multi-core certification Job-Aid for AMC 20-193

James Sharp [a], Mike Standish [a,*], Jaspal Sagoo [b], Edwin van de Sluis [c]

[a] Dstl, Fareham, UK
[b] QinetiQ, Malvern, UK
[c] Royal NLR, Amsterdam, The Netherlands

ARTICLE INFO

Keywords:
Multi-core Processors (MCPs)
Electronic hardware
Software
Safety-critical
High-criticality
Certification
FAA
EASA
Job Aid
Aviation
Aerospace
Civil
Defence

ABSTRACT

Multi-Core Processors (MCPs) are ubiquitous in modern electronic devices. However, their exploitation within the high criticality domains, specifically that of aerospace, introduces challenges. The European Union Aviation Safety Agency (EASA) and the Federal Aviation Administration (FAA) recently released harmonised guidance in the form of Acceptable Means of Compliance (AMC) 20-193, which details *what* is required, from a certification perspective, to enable the use of MCPs for satisfying airworthiness requirements. Although regulatory authorities have withdrawn Job Aids for standards such as DO-178 and DO-254, they are an effective method of showing compliance to standards and widely used by assessors. Understanding MCPs is, however, non-trivial and requires significant expertise not only of the device itself, but also how software will be architected and executed, along with how system level safety considerations are to be employed, all to ensure safe application of this technology. Thus, within this paper the authors, through the provision of an assessment of the *what* detailed in AMC 20-193, give an in-depth analysis into the intent behind the 10 objectives set out in this new AMC. The aim of the paper is to provide a foundation upon which Subject Matter Experts (SMEs) might construct their own Job Aid. Through its discussions, it is the authors intention that this paper enables a common understanding against which an applicant, assessor, and authority can interpret the *how* when looking to achieve the *what* set out in AMC 20-193.

1. Introduction

To ensure safety, high-criticality domains (such as aerospace) undertake significant scrutiny of the underpinning Airborne Electronic Hardware (AEH) and the associated software. In civil aviation, the use of software and AEH in practice requires the application of standards such as DO-178C [1] and DO-254 [2], respectively, to ensure safety considerations are met.

Whilst the use of AEH and software has become more prevalent in the aerospace domain, particularly in defence, this domain is no longer the key consumer of electronic components [3]. Instead, other industries such as automotive, mobile devices, and data centres are the market drivers; particularly for processors [4].

Since the aerospace domain is no longer the driver for the semiconductor industry, it therefore has limited influence and choice over the available devices. For instance, considering that Multi-Core Processors (MCPs) have been in circulation for decades (being first introduced by International Business Machines Corporation (IBM) in 2001 [5]), their application in the aerospace domain has been significantly delayed. Indeed, it is only recently that MCPs have been introduced in air platforms [6]. A key factor for this delay is the risk of increased non-determinism introduced through the use of shared resources by the multiple cores within an MCP. That is, the increased unpredictability in computational time for hard real-time tasks (as a result of this non-determinism) is undesirable for safety-critical functions. Methods of bounding and quantifying this unpredictability have therefore been required.

To that end, there have been multiple academic research activities (e.g. the University of York [7], Carnegie Mellon University (CMU) Software Engineering Institute (SEI) [8], and Imperial College London [9]) into understanding how to quantify and mitigate the additional complexities introduced by MCPs, most of which target the avionics and high-criticality domains. Industrial agencies have also undertaken significant research into approaches and tooling to enable the use of MCP devices in aviation (e.g. Airbus [10,11], nHansa [12,13], Thales [14,15], and Rapita Systems [16]), and has even been a focus of civil aviation authority funding (e.g. DGAC's PHYLOG project[1]). In January 2022 European Union Aviation Safety Agency (EASA) released Acceptable Means of Compliance (AMC) 20-193 [17], providing a set of objectives

* Corresponding author.
E-mail address: mstandish@dstl.gov.uk (M. Standish).
[1] https://hal.science/PHYLOG.

https://doi.org/10.1016/j.csi.2025.104049
Received 12 June 2024; Received in revised form 30 May 2025; Accepted 24 July 2025
Available online 5 August 2025

to support the use of MCPs in aviation's most safety-critical cases (up to Item Design Assurance Level (IDAL) A).

AMC 20-193 is a refinement of CAST-32A [18], which has been scrutinised by academia, industry and regulatory bodies. The objectives of AMC 20-193[2] require fulfilment through the generation of artefacts that detail both a strategy toward certification and evidence of its execution. However, the AMC details only the 'what' in these objectives, not the 'how'; as was the intent of the AMC authors [19].

This paper examines the objectives of AMC 20-193, with the aim of providing additional guidance to fulfilling these objectives. The intention is to propose guidance that can be used to inform the construction of an MCP Job Aid to support adherence to AMC 20-193. Moreover, this guidance can assist in the process of producing Job Aids, which can act as frameworks (e.g. [20]) to manage the inputs, transition criteria, planning, execution, outputs, and completion of formal review processes.[3]

1.1. Methodology

In forming guidance for an MCP Job Aid, the authors identify two questions that are considered to be key in addressing certification against AMC 20-193:

Q1. *What structure can be added to the objectives, adding clarity as to when they need to be addressed within a development lifecycle?*

Q2. *What suitable questions can be identified that add context and harmonised interpretation, between regulator and applicant, to initiate and guide discussion when addressing the AMC's objectives?*

The methodology adopted to address Q1 and Q2 was based on conducting a series of technical workshops, which is a key qualitative research method that is widely used within industry and academia. In order to increase the robustness of our work, in addressing Q1 and Q2, the workshops used a framework that combined:

- Primary data, which was based on the experience and knowledge of the authors as Subject Matter Experts (SMEs), in conducting assurance assessments and audits of AEH and software on various military projects. Since the authors have also advised industrial and government organisations on the use of standards (such as DO-254 and DO-178) for certification, this knowledge was used to interpret the AMC objectives.
- Secondary data, which used accessible publications to support the brainstorming discussions held during the workshops. It should be noted that although research into MCPs contains a plethora of publications, this paper does not provide an exhaustive (or comprehensive) literature review of MCPs, as the authors view this not to be within the goals of this paper. The workshops used relevant publications that allowed the authors to gain an insight into understanding Q1 and Q2, and forming relevant guidance.

The workshops were structured into the format of:

- For each AMC objective, considering its meaning, interpretation and intent from the perspective of the assessor and supplier.
- Determining what additional questions should be posed and activities performed to satisfy the AMC objectives. This stage used both primary and secondary data sources but heavily relied on the former (i.e. the authors expertise of actually performing Stage of Involvement (SOI) reviews for the certification of AEH and software).
- Gaining a consensus amongst the workshop participants on the type of guidance that should be provided.

The output of these workshop sessions (which forms this paper) provides: a multi-SME agreed interpretation of the AMC objectives; an understanding of when and how each objective should be addressed within a development lifecycle; and a set of supporting questions to illicit the creation of suitable responses to meet the overall intent of the AMC, and to address each AMC objective. The authors believe that the work presented in this paper, underpinned by workshops and the experiences of SMEs, provides a meaningful and relevant contribution to the ongoing practical efforts for applying AMC 20-193.

1.2. Structure

Section 2 discusses the concept of Job Aids and their use in civil regulation. In Section 3 we introduce the certification standards used for software/AEH within aircraft and how AMC 20-193 sits within a civil certification approach.

In Section 4, we evaluate each of the 10 objectives introduced through AMC 20-193.[4] From this evaluation, we introduce additional questions to drive a deeper granularity of questioning, along with identification of linked activities to support the fulfilment of the objectives. The detail provided in Section 4 could form the basis of a Job Aid for any personnel performing technical evaluations as part of a certification process. In addition, Section 4 provides supportive material for applicants adopting MCP technologies, motivating them to generate suitable, consistent, and co-ordinated artefacts.

In Section 5 we provide an evaluation of the work conducted, and wider considerations towards the certification and use of MCPs. Finally, in Section 6 we provide a conclusion, a set of recommendations, and details of potential next steps for the work.

2. Job Aids: An Introduction

2.1. Job Aids for DO-254 and DO-178C reviews

Within the aviation domain, a typical DO-254/DO-178 certification assessment of AEH and software would undergo four SOIs[5] [22,23]. These stages are identified as follows:

- SOI #1 Planning Review:

  "A … planning review should be conducted when the initial … planning process is complete (i.e. when most of the plans and standards are complete and reviewed)".

- SOI #2 Development Review:

  "A … development review should be conducted when all actions from the … planning review (SOI #1) have been proposed for closure and at least 75% of the … development data (i.e. requirements, design and code) are complete and reviewed".

- SOI #3 Verification Review:

  "A … verification review should be conducted when at least 75% of the … verification and testing data are complete and reviewed".

- SOI #4 Final Certification Review:

[2] Note that the terms 'AMC' and 'AMC 20-193' are used interchangeably throughout this paper.

[3] Further information is contained in Section 2.

[4] This includes MCP_Resource_Usage_2, which references out to AMC 20-152A [21].

[5] The term SOI was first mentioned in the now withdrawn software Job Aid [22], and later used within the Federal Aviation Administration (FAA) AEH job aid [23]. Note that EASA [24] instead used the terms 'Hardware Reviews', but they are suitably similar.

> *"A final certification … review should be conducted after the final … build is completed, the … verification is completed, a … conformity review has been conducted, and the … product is ready for formal system approval".*

When undertaking SOIs, the FAA have previously employed the use of 'Job Aids' to support the assessment at the various SOIs. Specifically, a Job Aid has traditionally been intended for use as an *aide-memoir* by the Designated Engineering Representative (DER), or their delegate. For instance, the DO-254 review Job Aid [23] stated[6]:

> *"This Job Aid should be used as a reference tool during the review process. It is not intended to be used as a checklist and is not all inclusive of all possible situations that need to be reviewed. Nor is the Job Aid intended to replace DO-254. Rather, it should be used in conjunction with DO-254".*

The Job Aids not only consist of SOI review stages, but also include the roles and responsibilities of those that attend the reviews, and how a SOI should be organised. The Job Aids are more than just a checklist of questions, they act as a framework to manage the inputs, transition criteria, estimated duration, planning, execution, outputs, and completion of the SOI process.[7]

It is important to note that a SOI may well be carried out before the associated stage of development is completed. For instance, SOI#2 may be enacted when the applicant identifies that they have completed 75% of the design stage.

2.2. Job Aids: Their value

Whilst DO-254 has until recently been supported by a Job Aid [23],[8] the FAA has now withdrawn the Job Aids for both AEH and software on the release of DO-178C. The rationale for the removal of the Job Aids is not completely clear, however since the Job Aid for DO-178B required updating (due to the additional objectives of DO-178C and its supplements), it became out-dated. Furthermore, certification has, generally, been regarded as a subjective activity and seen by some as a 'tick box' process [25]. Since the success of an AEH or software compliance activity is dependent on the experience and skills of the assessor, authorities (such as the FAA) may have wanted to encourage assessors to form their own means of showing compliance.

The withdrawal of these FAA Job Aids has not, however, meant that checklists (such as [26] for DO-178C) have stopped being produced (or used) nor has it diminished the appetite for Job Aids. In fact the use of Job Aids forms an important means of showing compliance to the certification objectives of DO-178C or DO-254, and ensuring the 'completeness' of an assessment [27].

It is important to recognise that whilst Job Aids may pose a risk if used by assessors as a 'tick box' exercise, they are still a powerful tool in supporting the assessment of AEH and software in aviation when used by experienced, knowledgeable, and careful assessors.

Based on the utility of Job Aids, in the remainder of this paper we propose to develop the basis of a Job Aid for AMC 20-193. This work is a culmination of a multi-national effort to develop a means to show compliance to AMC 20-193. As MCPs are still a novel technology for certification within aviation, the authors consider this work to be a useful addition to the certification process that uses AMC 20-193.

Importantly, unlike a traditional software or AEH SOI assessment, when MCPs are adopted, there is a need to consistently and coherently employ Suitably Qualified and Experienced Personnel (SQEP)[9] SMEs for systems, hardware, and software when addressing the use of MCPs. This is to ensure that the full considerations of the AMC are addressed.

3. Framing AMC 20-193 and its related standards

3.1. AMC 20-193: Not the only approach

This paper does not critique or necessarily 'promote' AMC 20-193. The paper acknowledges that AMC 20-193 is a widespread means of compliance to demonstrate MCP assurance. Indeed, for the civil sector AMC 20-193 has become the *de facto* means of compliance, despite guidance existing within the defence domain, such as Airworthiness Advisory (AA) 22-01[10] to MIL-HDBK 516C [29] and the United States (US) Army Software Airworthiness Qualification Requirements for Multi-Core Processors [30].

The paper authors cannot state that the information in the Job Aid (outlined in this paper) would meet the requirements/intent of other MCP guidance but there are strong similarities between the AMC and other defence guidance. Therefore, although not categorically written to support other guidelines, an AMC 20-193 Job Aid could have wider MCP assurance applicability.

3.2. AMC 20-193 and the wider regulatory requirements

From a civil perspective, software/AEH of interest (e.g. due to safety or complexity) are commonly assessed against DO-178C [1] for software and DO-254 [2] for Complex Electronic Hardware (CEH).[11] These guidelines have pedigree and have been in active use for decades, indeed DO-178 was originally developed in the late 1970s with a number of iterations (DO-178A in 1985 and DO-178B in 1992) to reach the current version (DO-178C) in 2012. DO-254 was released in 2000 and 'formally' adopted by the FAA in 2005.[12]

A benefit of DO-178C and DO-254 is that they are predominantly process-based and are life-cycle and technology agnostic. However, as a consequence, systems may be implemented with technologies/features (which have an impact on safety) that are not *explicitly* covered by existing guidelines. That is, technologies now exist that were not considered, or indeed did not exist, at their time of their writing, and as such these two standards may not be sufficient on their own.

One such technology is the Multi-Core Processor. The concerns of using MCPs are typically related to aspects on interfacing of software with the hardware over and above that generally considered under DO-178C's "compatibility with target computer" objectives. In 2014, the Certification Authorities Software Team (CAST) released information (but not official guidance — it is not within the CAST remit) on how to potentially allow the use of MCPs in the civil domain (the released paper was called CAST-32 [33]). The MCP CAST-32 paper was updated to CAST-32A [18] in 2016.

The CAST-32A paper presented 10 objectives for Design Assurance Levels (DALs) A and B, with 6 for DAL C; these are based upon planning, resource usage, software, error handling, and an accomplishment

[6] The DO-178B Job Aid contained similar text but with reference to DO-178 rather than DO-254.

[7] Note that it is not the intent of this paper to address all of these points, for instance the estimated duration for life-cycle stages is not considered.

[8] The DO-254 Job Aid has recently been withdrawn from the public domain, circa mid-2023. It is the authors' belief that the release of AC 20-152A by the FAA has triggered the removal of what is now considered an outdated DO-254 Job Aid.

[9] The term *SQEP* is defined as a designated individual who – by virtue of their training, experience, recency in role and personal characteristics – is expected to be competent to fulfil a specified role. This individual will be appropriately empowered to act within the context of their SQEP responsibilities [28].

[10] Noting that AA 22-01 still has a draft status.

[11] It should be noted that DO-254 uses the term 'CEH and EASA use the term 'AEH'. In this paper the terms are used interchangeably.

[12] DO-178 and DO-254 are enacted within the civil domain through AC 20-115 [31] and AC 20-152 [32].

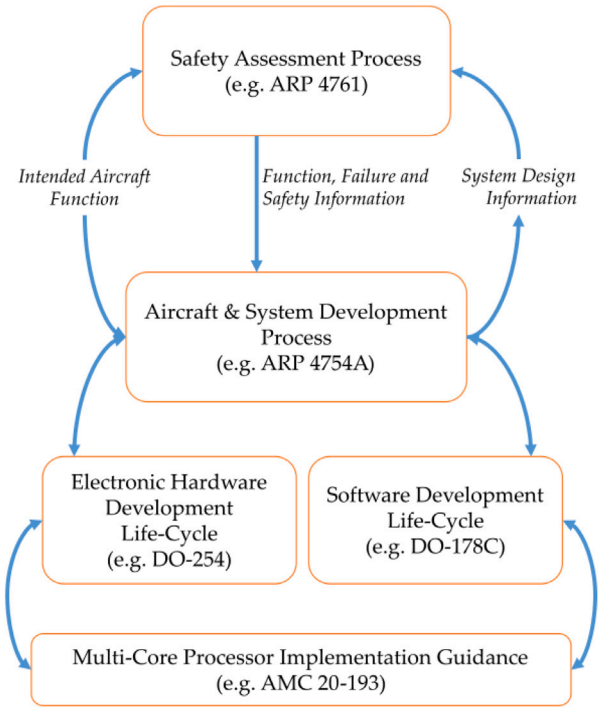

Fig. 1. Wider regulatory framework to MCP implementation guidance.

summary.[13] These objectives contain a number of guidance notes which in effect act as 'sub-objectives'.

AMC 20-193 further refines CAST-32A but, importantly, provides no change in position or the applicability to DALs. The rationale/context for each of the objectives is equivalent (although less verbose), but with some additional guidance notes. The need for hardware considerations when certifying the use of MCP, and thus the relationship to DO-254 is emphasised within AMC 20-193 as MCP_Resource_Usage_2 directly references the recently released AMC 20-152A [21],[14] Objective COTS-8. The COTS-8 objective was previously contained within CAST-32A.

Traditionally, a safety assessment process (e.g. ARP 4761 [34])/ system development process (e.g. ARP 4754A [35]) would allocate appropriate software/AEH DALs. The DALs placed on the MCP implementation, would consequently be allocated from these DALs, with the architecture developed accordingly (including the use of independent functions). The flow from the safety assessment process to the MCP objectives is simplistically illustrated in Fig. 1.

Fig. 2 visualises the approximate timeline at which relevant standards/guidelines were introduced. This Figure also shows the timeframe (approximately 20 years) between the first introduction of MCPs and their use within airborne platforms. Moreover, Fig. 2 highlights the relevant disparity between the release of standards/guidelines with respect to each other and the resultant potential discontinuity. Finally, and perhaps most importantly, the Figure demonstrates the protracted and conservative incremental development of guidance (educational and the resultant AMC) against new technologies, such as MCP.

AMC 20-193 is not only starting to gain acceptance within the civil domains, but also defence regulators (such as the United Kingdom (UK) Military Aviation Authority (MAA)) are now expecting the AMC to be adopted for Type Design Changes which involve MCPs [36]. The UK MAA process for MCP assurance (using the guidance in AMC 20-193) is to develop a Military Certification Review Item (MCRI) which allows the approach to be articulated to the authority (as set out in [36]). The use of safety-critical MCPs within the UK defence domain is still viewed as 'novel', despite the formal release of AMC 20-193. This indicates that the adoption of MCPs still pose technical and assurance challenges. Educational outputs (such as Job Aids) would, we believe, therefore be a useful addition to the assurance domains (both civil and defence).

4. Expanding AMC 20-193 objectives

Within this section we evaluate the intent of each objective, and identify where within an SOI engagement, one might expect to see artefacts to support its achievement. Note that for some of these objectives, activities should be undertaken iteratively through the SOIs.

The following provides a discussion (which is based upon the authors' understanding/interpretation) against each of the objectives accompanied by a set of questions to support the production of a tailored Job Aid. It should be noted however, that the information provided is for guidance only and should *not* be considered the *de facto* baseline. These may differ, dependent on the assessor's own position.

4.1. MCP_Planning_1

The first MCP objective in AMC 20-193 is partitioned into 7 specific sub-objectives that should be addressed (and documented) as part of the planning phase. Specifically, they deal with the proposed MCP device and how it is intended to be used. Considerations regarding these 7 sub-objectives are expanded below, where deemed appropriate.

4.1.1. MCP_Planning_1.1: Identify the specific MCP, including the unique identifier from the manufacturer

Discussion

A chosen MCP should be as a result of an informed decision and the rationale behind the device selection should be exposed. This may be a simple statement such as: "Our engineers have significant previous experience with this device family, it provides sufficient features for the functionality required, and there is already an Non-Disclosure Agreement (NDA) in place with the device vendor". Conversely, a rationale for the down-selection involving a detailed analysis of multiple device vendors or even device models may need to be considered. Particularly if there are specific device features intended for use that may incur certification challenges (as per MCP_Planning_1.4).

Along with the identification of a proposed device (including the make, model, and variant), it is advantageous that characterisation/ profiling [9,13] be undertaken and made available. This is of particular benefit if the results of this device profiling informs the overall certification approach; such as informing on the usage of an Integrated Modular Architecture (IMA) argument [37].

Additionally, non-technical considerations should also be addressed. For instance, if a device is being chosen due to the availability of an AEH MCP certification pack, the certification pack is likely to be provided under an NDA. Thus, the applicant should consider provision of sufficient information to their regulator/assessor, such that the NDA either includes the regulator, or sufficient information provided within a certification argument is not in violation of the NDA.

SOI#1 questions

1. Are the specific MCP processors identified, including the unique identifier (such as the part number, make, model and variant) from the manufacturer?
2. What is the rationale for the down-selection of the processor device?
3. What characterisation or profiling of the device(s) informed selection is being used and where is this documented?
4. Is characterisation available or planned, and if so, what is the confidence that the MCP will be suitable?

[13] For brevity, these topics will not be expanded further in this section but will be covered in detail within the rest of the paper.

[14] Released in conjunction with AMC 20-193.

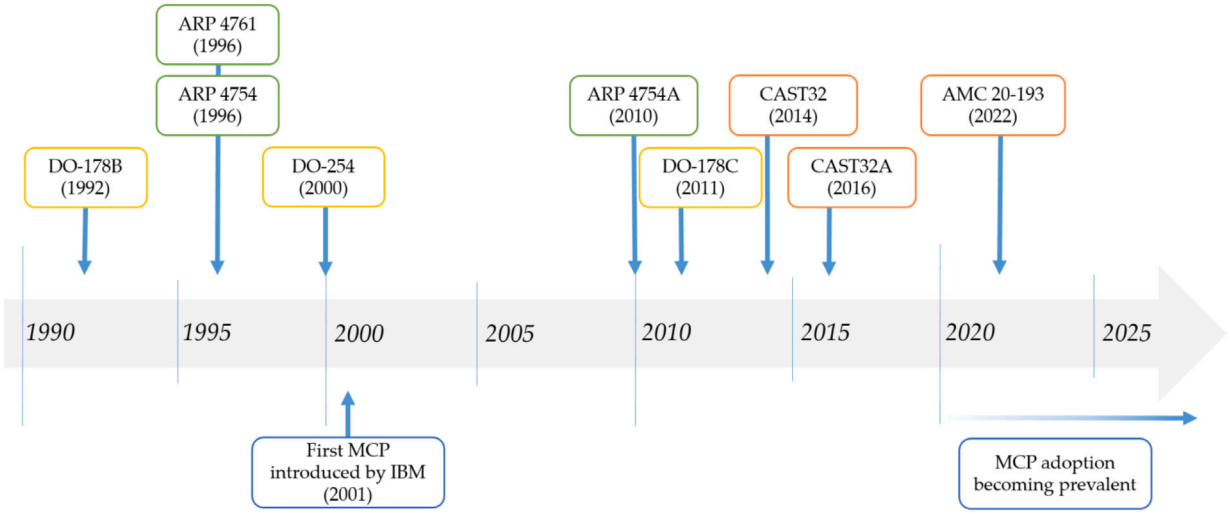

Fig. 2. Timeline of selected standards/guidelines.

5. Does the planned MCP have a suitable AEH MCP certification package? If so, are these packages available for review and free of NDA access conflicts?

4.1.2. MCP_Planning_1.2: Identify the number of active cores

Discussion

When an MCP is intended for use with disabled cores, a strategy for managing the unused cores should be employed. Some vendors may provide methods for physically turning off the devices through routing of power (or lack thereof). Alternatively, an MCP device may offer the ability to disable cores within a tailored Basic Input Output System (BIOS). Consideration should be taken as to whether this is a viable option, or if holding cores within a spinlock [38] is preferable. Indeed, some device vendors provide libraries to support such methods [39]. Again, evidence of the sufficiency of any chosen approach should be addressed as part of the characterisation/profiling of the device. If this configuration management occurs within software, then the DO-178C objectives related to Parameter Data Item (PDI) may also need be considered.

Further to the identification of active cores, assessment of the device needs to include understanding whether core bias exists [40] (where a core(s) has preferential access to shared resources), and if so how this will be mitigated. It may be that core bias information is provided within vendor documentation. Equally, such a bias may only be discovered as a result of device characterisation/profiling. And, depending on the vendor, the method of arbitration in the enumeration and placement of cores within the chosen MCP may need to be considered.

The determination of specific active cores may also be important where multiple Memory Management Units (MMUs) within the MCP are intended to be used, offering a reduction in potential interference paths. Such functionality may be provided within documentation, or may require direct engagement with the MCP vendor.

Once the impact of core usage/disablement is fully understood, and the proposed configuration of core(s) is determined and justified, it should be suitably documented. Further, consideration must be made to ensure that the configuration of the used/unused core(s) is secured, or, in the event of an architectural update, revalidated.

Finally, depending on the certification strategy being undertaken, the monitoring of active and deactivated cores should be considered. This monitoring would be necessary to ensure that the cores remain in their intended states and to avoid inadvertent functioning. Such monitors may be addressed in response to fulfilment of MCP_Error_Handling_1.

SOI#1 questions

1. What are the number of active cores and are they clearly identified?
2. If the processor has been profiled (core biasing), have the cores been enumerated?
3. If cores are unused within the design: what is the deactivation method (for cores not used within the design); how will this be monitored, controlled, and measured (if applicable) during use; and how will this be verified?
4. How is each core configured and how is this configuration managed?
5. How is the configuration secured/managed with respect to factors such as changes/updates?

4.1.3. MCP_Planning_1.3: Identify the MCP software architecture to be used and all the software components that will be hosted on the MCP

Discussion

In addition to the sub-objectives, AMC 20-193 includes notes associated with MCP_Planning_1. The first of these notes applies here:

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

It is not the intent of this paper to detail the differences between AMP, SMP, or even bound multi-processing (BMP).[15] It is anticipated (based on the authors' experience at the time of writing) that the majority of applicants will propose the use of level 1 hypervisors[16] along with an AMP approach.[17] This enables exploitation of the SWaP benefits of MCPs, along with re-hosting of existing software. This approach does have the potential to limit true exploitation of the MCPs, as assigning individual cores to separate Real-Time Operating System (RTOS) instantiations, may not support a truly parallelised compute approach. Applicants may transition toward BMP based solutions, as

[15] A summary of the differences between AMP, SMP and BMP may be found in [41].

[16] Further information on the types of hypervisors can be found within [42].

[17] An AMP approach utilises multiple cores, potentially with differing architectures, often with the option for each to have individual address spaces, may (or may not) run differing Operating Systems (OSs), and can enable mixed-criticality; thus, offering maximal Size Weight and Power (SWaP) benefits.

the need for higher throughput and parallelised processing is employed, and as greater understanding in handling interference paths when utilising methods such as core affinity is achieved. Further, BMP may be necessary where mode changes are required during operational flight, requiring associated schedulers to maintain safety, such as [43].

In addition to the MCP software architecture, there is, as the sub-objective states, a need to detail all the software components that will be hosted; thus, the second note is also applicable:

> *"(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".*

It is therefore advisable to compile a complete list (or as complete as possible) of all software components in support of MCP_Software_1 as early as possible within the life-cycle.[18] Furthermore, for any procured software, such as RTOSs and hypervisors, early identification of their 'certifiability status' is considered advantageous. The configuration of applications to be hosted on the MCP may play a part in the chosen RTOS and/or hypervisor. Decisions may be driven by how flexible the architecture needs to be: can it be configured using PDIs, does it support dynamic relocation, or will hardware features such as Processor/Core Affinity [44,45] be employed? Again, rationale and engineering judgement presented at this stage has the potential to minimise challenges/re-work later in certification.

SOI#1 questions

1. What is the proposed MCP architecture and is it clearly identified? The expected terminology is "AMP, SMP, and BMP".
2. Are (all) software components that will be hosted on the MCPs identified? *Note: Software components include any part of the software which may access MCP shared resources: software applications, OSs, hypervisors, or Board Support Package (BSP)/drivers.*
3. If RTOS/OS are planned to be used, what is their 'certifiability status'?
4. If hypervisors are planned to be used, what is their "certifiability status"?
5. How are applications intended to be configured for the platform? *Note: the software architecture configuration method (e.g. PDI) and leveraging of hardware features should be identified.*

4.1.4. MCP_Planning_1.4: 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

Discussion

AMC 20-193 includes notes associated with MCP_Planning_1 which can be directed toward this sub-objective:

> *"(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".*

In addition, section 2.2.1 of the AMC also stipulates:

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

Considering the AMC, along with the note, it is clear that a corporate OS approach (such as Windows 11) to process allocation is not within the scope of the AMC as it is determined and managed at runtime. However, a pre-defined process allocation that is verified (as part of MCP_Resource_4 activities), and captured within the certification artefacts, is permissible. Further, the dynamic allocation of RTOSs/OSs to different cores upon start-up is not considered within the bounds of the AMC unless, again, this is part of the supplied certification argument.

"*Dynamic features provided in software*" is also read to mean spatial allocations such as how software may allocate memory or peripherals (such as serial and Ethernet, etc.). Temporal dynamic features provided by software must also be considered, such as cases where schedulers are employed within an RTOS (e.g. rate monotonic). The certifiability of any dynamic software features employed, be they Commercial-Off-The-Shelf (COTS) or bespoke, needs to be addressed.

SOI#1 questions

1. Are any dynamic features provided in software hosted on the MCP identified, and if so is a (high-level) description of the activation mechanism and purpose provided? *Note: it is anticipated that significant inquiry may be required between applicant and Design Authority wherever dynamic allocation is employed.*
2. If a rate monotonic scheduling solution is to be employed, is it a bespoke or off-the-shelf solution, and is it certifiable?

4.1.5. MCP_Planning_1.5: Identify whether or not the MCP will be used in an IMA platform to host software applications from more than one system

Discussion

The intent of this paper is not to explain IMA, however, for the purposes of this paper it is noted that IMA enables an incremental certification approach for a platform to be undertaken. The intent being that hosted applications can be independently verified, and credit towards their approval achieved, prior to integration. Whilst is it not required under the standard for IMA (i.e. DO-297 [37,46][19]) to identify all the software applications that are going onto the platform, AMC 20-193 introduces a specific restriction with regards to IMA:

> *"this term refers to an integrated modular avionics MCP platform that provides both robust resource partitioning and robust time partitioning (as defined in this AMC)".*

Thus, the use of an IMA in the context of AMC 20-193 would mean that the first of the two approaches to MCP_Software_1, MCP platforms with robust partitioning, would be taken.

The rationale, when using an IMA on an MCP, should lead the applicant at the early stages to identify any hypervisors and the RTOSs/OSs intended for use (that support temporal and spatial partitioning). These constraints should be captured within the responses to MCP_Planning_1.3. Given that these constraints are met, in line with DO-297, identification of the 'hosted' software applications may be postponed to a later stage of the certification process, leveraging the IMA principles, and alleviating the need to support MCP_Planning_1.3 to the fullest extent.

SOI#1 questions

See MCP_Planning_1.3.

[18] Noting that such an activity is to achieve compliance to the AMC, it is acknowledged that the architecture will likely evolve throughout the development. Therefore, such changes should be tracked and assessed appropriately.

[19] Additional guidance regarding DO-297 is provided in AMC 20-170.

4.1.6. MCP_Planning_1.6: Identify whether or not the MCP platform will provide robust resource partitioning and/or robust time partitioning as defined in this AMC

Discussion

The AMC 20-193 Definition for robust resource partitioning is given as being achieved when:

- *"software partitions cannot contaminate the storage areas for the code, Input/Output (I/O), or data of other partitions* (spatial)*;*
- *software partitions cannot consume more than their allocations of shared resources* (spatial and temporal)*; and*
- *failures of hardware unique to a software partition cannot cause adverse effects on other software partitions* (mitigation of a side effect)*".*

The above definitions express that there is a need to identify whether or not the MCP platform will provide robust temporal partitioning. This is achieved when, as a result of mitigating temporal 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. Moreover, this is irrespective of whether partitions are executing on none of the other active cores or on all of the other active cores.

In addition to fulfilment of robust temporal partitioning, it is also reasonable that failure detection/mitigation, along with recovery plans/methods be considered. However, these should be addressed when answering MCP_Error_Handling_1.

To sufficiently answer this objective, there is an implicit requirement to have answered the prior sub-objectives. Specifically, questions around: the choice of AMP, SMP, or BMP; the use of dynamic features provided in software; and whether an IMA is being employed. In answering these earlier sub-objectives, it is expected that a clear rationale for a robust partitioning in either or both resource and time is to be accomplished.

It should be recognised that the viability of a temporally and spatially robust partitioning system for MCPs has been called into question in recent years.[20]

SOI#1 questions

Note: some of the questions provided below are repetitions, but must be answered for the achievement of this sub-objective.

1. What is the intended MCP software architecture?
2. For the architecture identified, what is the approach used for providing robust resource partitioning?
3. For the architecture identified, what is the approach used for providing robust temporal partitioning?
4. What are the main causes for temporal interference by/between cores? How will they be identified?
5. What schedulability approach will be applied, and how is this planned to be assessed at the core and processor level?
6. If a robust time and resource partitioning approach is being taken, has a credible approach been established? *Note: such an artefact would also be used for the satisfaction of MCP_Software_1.*

4.1.7. MCP_Planning_1.7: 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 AMC and the applicable software guidance, including any methods or tools needed due to the use of an MCP or the selected MCP architecture

Discussion

The use of tools is ubiquitous within the development of embedded software systems. Ignoring tools such as compilers and other non-MCP specific tools (which are considered under DO-178C), there are a variety of tools that may be used to support the development and, more specifically support the certification of the use of an MCP. It would be the responsibility of the applicant to determine where tools may be used to support any certification argument made. Common examples may include those that: support profiling of the MCP and/or generate and configure demonic processes[21] [48] to stress test the software architecture. Any, and all tools, that provide certification credit (or derive certification evidence) should meet the requirements of standards such as DO-330[22] [49] (or equivalent).

SOI#1 questions

1. What are the methods and tools identified that are to be used to develop and verify all the individual software components hosted on the MCP?
2. What kind of tool support is available for critical configuration settings?
3. What methods and tools are identified due to the use of an MCP, e.g. to support interference mitigations?
4. Which software development and verification tools are planned to be used?
5. Which of the identified tools require qualification and why? *Note: by implication, the software tools typically need to be classified as per DO-178C and shown to be compliant (or equivalent) to DO-330. A similar approach should be taken for AEH tools.*

4.2. MCP_Planning_2

MCP_Planning_2 is partitioned into 4 specific sub-objectives that should be addressed (and documented) as part of the planning phase. Specifically, they concern the proposed MCP device and how partitioning of its shared resources are to be used.

4.2.1. MCP_Planning_2.1 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) so as 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

Discussion

Hypervisors (and indeed separation kernels) may leverage dynamic hardware functionality within an MCP device through features provided within the software [50]. For instance, exploiting the hardware specification single root I/O virtualization (SR-IOV) [51] to create multiple virtual instantiations of a network interface presented to the hypervisor, one for each active core of the MCP, against a single physical interface. Similarly, many MCP devices enable hardware partitioning of caches and memory [52]. It is these sorts of features which should be considered when responding to this sub-objective. Indeed, the following notes are provided within the AMC that are applicable for this sub-objective:

> *"(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*

[20] For example, see https://www.lynx.com/embedded-systems-learning-center/robust-partitioning-is-dead-what-now.

[21] The origins of demonic processes are not derived from the exploitation of MCPs within safety-critical systems, but have become more prevalent in response to their introduction, as witnessed by their 'namesake' being employed in supportive tooling such as RapiDaemons [47].

[22] Whilst AMC 20-193 does identify DO-330 within its reference section, it is not explicitly referenced within the main body.

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.

The fulfilment of this sub-objective can, from the authors' view, be met through the introduction of a *life-cycle strategy for interference analysis*. Not only would such an artefact provide detail on the types of interference that require mitigation in the device, but also detail what dynamic hardware functionality may be used and thus needs to be considered. Further, such an artefact, as the name suggests, would enable considerations on what and when different analyses should be performed. It would also be a living document throughout the development life-cycle and thus could be updated and cross-checked through to completion of certification.

As previously mentioned within MCP_Planning_1.2, MCP devices may contain core bias, which can be implicit as a result of the physical properties of the chip. For instance, the proximity of a core to the MMU may result in a biasing, or there may be an implicit arbitration within the chips design between different aspects of the device (e.g. Peripheral Component Interconnect Express (PCIe) vs Ethernet). A possible example of interference considerations is provided by the US Army MCP Interference Matrix [53].

It is noted, by the authors, that many modern MCP devices offer simultaneous multithreading (SMT) (e.g. Intel's hyper-threading), which utilises unused stages within each core's pipeline to effectively provide a virtual core. Decisions on the spare capacity of a cores pipeline is undertaken in hardware, and may be highly non-deterministic. Possibly due to the challenges associated with SMT, section 2.2.2 of AMC 20-193 states that "*this AMC does not cover simultaneous multithreading, as this issue is not specific to MCPs*".

SOI#1 questions

1. Is there a high-level description of how MCP shared hardware resources will be used?
2. How does the applicant intend to allocate and verify the use of shared resources, such as including the use of shared:
   - caches, e.g. L2 and L3,
   - external (to the MCP) memory,
   - interconnect, e.g. internal buses or logic,
   - I/O, e.g. PCIe, Ethernet, Universal Asynchronous Receiver-Transmitter (UART), Inter-Integrated Circuit (I2C)?
3. What are the foreseen interferences due to arbitration for access to the shared resources? *Note: This is about understanding the constraints/behaviour of the targeted MCP, and may require initial analysis to support any argument made.*
4. Do the plans include a process for re-evaluation of interference channels in line with MCP_Resource_Usage_3 (see MCP_Resource_Usage_3 Note c)?

4.2.2. MCP_Planning_2.2: Identify the MCP hardware resources to be used to support the objectives in this AMC

Discussion

Thus far, objectives have focussed on identifying possible areas of non-determinism, and how these will be controlled or associated features deactivated. However, for this sub-objective, the applicant is expected to highlight any and all aspects of a device that are leveraged of the hardware device to eliminate, mitigate, or control sources of interference. These may be hardware controls for memory isolation and core partitioning (e.g. Intel's VMX/VT-x), virtualisation methods to provide schedulability over shared resources (such as SR-IOV), or leveraging of internal device registers to identify device state (e.g. core activations).

It is unlikely that the requisite detail required for certification for such devices will be evident without complete access to the device's data sheets, or highlighted within certification packs. Thus, there is a need for suitable data access agreements to be in place, e.g. via NDAs; not only for the applicant but also potentially for the assessors.

SOI#1 questions

1. What are the MCP hardware resources that are to be used to support the objectives in this AMC? *Note: the hardware resources could be Instruction Set Architecture (ISA)/device configuration hardware acceleration elements used to support resource constraints e.g. Intel VT-x, or registers used for verification of cache misses, for example.*
2. Are the hardware resources identified from sources such as the device's data sheet? Has their role in supporting the relevant objectives been identified/mapped? *Note: the use of some hardware resources may require the exclusion of other hardware resources.*

4.2.3. MCP_Planning_2.3: Identify any hardware dynamic features of the MCP that will be active, and provide a high-level description of how they will be used

Discussion

The complexity of modern processors may introduce internal monitors that can control dynamic features that are not only leveraged to improve energy efficiency, but may also be used to protect the devices from damage. An example of this is Intel's Management Engine (ME) [54], which is able to reduce the frequency of a core, or indeed an entire MCP device, to reduce thermal output when dangerous levels are detected without interacting with any software. For a desktop or data centre, this is considered a beneficial feature. However, for safety-critical real-time operations such device features need to be identified and assessed for suitability.[23]

Whilst many of the dynamic features of the hardware may need to be mitigated, there may be some that are leveraged as part of a system implementation. For instance, different modes of use (such as for operation or maintenance) may require different dynamic configurations (enabling/disabling of various I/O or cores).

Thus, whilst a pre-defined configuration may be selected, the additional features, designed for the safety of the device, must also be considered. Specifically, the impact of these features should be assessed so that they are eliminated, mitigated, or controlled.

SOI#1 questions

1. Are there any hardware dynamic features of the MCP that will be active? Is a high-level description of how they will be used provided? *Note b of MCP_Planning_2 states: "hardware dynamic features of the MCP include any features that can alter the behaviour 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)".*
2. Have proprietary features, such as the Intel ME, been identified and assessed for impact on the application? Have these features been eliminated, controlled, or mitigated (e.g. Intel ME frequency throttling being sufficiently mitigated).

4.2.4. MCP_Planning_2.4: 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

Discussion

AMC 20-193 glossary includes a definition of 'safety nets' as:

[23] An NDA maybe required to access this level of detail.

"the employment of mitigations and/or protections at the appropriate level of aircraft and system design as a means 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 by nature independent of the source of misbehaviour. The safety net can include passive monitoring functions, active fault avoidance functions, and control functions for effective recovery of system operations from anomalous events".

The intent of this sub-objective stems from the considerations of MCP_Planning_2.2 and MCP_Planning_2.3, and highlights that an MCP device may not necessarily be able to manage or mitigate failures itself. Indeed, the fulfilment of this sub-objective should drive the compilation of considerations to be managed within MCP_Error_Handling_1. For instance, it may not be possible for an MCP device to safely restart a frozen or run-away core, but it may provide a set of output registers detailing core state (that can be monitored through an external watchdog, e.g. [11]). Thus, the output of this objective should be the identification of considerations requiring either a secondary, external safety device, or where specific device features may need to be employed in isolation to the rest of the device to ensure safety. *Note: Such considerations may require the involvement of System Safety.*

SOI#1 questions

1. What is the functionality of the MCP which operates in an undesirable way (such as over temperature, frequency throttling, Single-Event Effect (SEE), unavailable interface or resource, or other generic features of the MCP)? *Note: this task may require the applicant to consider a risk-based approach and an examination of the system-level fault analysis. Any considerations should be fed as an input into MCP_Error_Handling_1.*
2. What aspects of use of the MCP may require a 'safety net' or other mechanisms to detect and handle failures in the MCP? What assessment will be performed to show that the proposed 'safety nets' can detect and handle failures *Note: This point is also related to MCP_Error_Handling_1.*

4.3. 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

Discussion

This objective is felt to be self-explanatory as its fulfilment is a natural consequence of completing MCP_Planning_1 and MCP_Planning_2 in conjunction with a justified design decision. For completeness, some additional SOI #2 questions have been included.

SOI#2 questions

1. Is the device and hypervisor/separation kernel sufficiently understood to ensure that all applicable configuration settings are correctly controlled?
2. Are the impacts of changes to the MCP, such as updates to microcode and configuration settings, considered and documented?
3. What verification activities will be applied to the applicable configuration settings? This should include any proprietary configurations provided by third-party vendors (e.g. customised boot-loaders).

SOI#2-3 questions

Note: As SOI#2 may be conducted prior to full completion of the design stage, some of these questions may not be fully addressed, and as such require re-review at SOI#3.

1. Has the applicant determined (to a reasonable degree of confidence for SOI #2) 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?
2. Have any deviations from the MCP_Planning_1 and MCP_Planning_2 objectives been recorded and justified?

4.4. MCP_Resource_Usage_2: Reserved. Covered by AMC 20-152A, Objective COTS-8

Discussion

Pro memori, from AMC 20-152A:

"Objective COTS-8: If the complex COTS device contributes to DAL A or B functions, the applicant should develop and verify a means that ensures an appropriate mitigation is specified in the event of any inadvertent alteration of the 'critical configuration settings' of the COTS device. Note: The mitigation means might be defined at the hardware, software, or system level, or a combination of these. The mitigation means may also be defined by the safety assessment process".

AMC 20-152A defines 'critical configuration settings' as:

"Those configuration settings that the applicant has determined to be necessary for the proper usage of the hardware, which, if inadvertently altered, could change the behavior of the COTS device, causing it to no longer fulfil its intended critical function".

For MCP hardware it is important to recognise that critical configuration settings include device aspects such as enabled/disabled I/O, cores, buses etc., many of which may be settable within BIOS and boot configurations. However, for modern processors (not just MCP devices) critical configuration settings should also consider processor microcode. The impact of changes to microcode was demonstrated with the identification of the Spectre side channel attack [55]. The security 'fix' introduced by microprocessor vendors was in the form of a microcode update, resulting in significant processing speed penalties as higher-level instructions were restructured to remove the attack vector utilising speculative execution. This clearly illustrates that microcode changes can have a significant impact to timings for hard real-time systems.

Depending on the chip vendor, there may be various considerations that should be taken into account when reviewing the critical configuration settings of the hardware. For example, complex and independent management functions operating on co-processors within the device, such as the Intel ME not only providing monitoring functionality but also a trusted computing base [56].

All of the above should be suitably captured and managed. It is also advisable to have a strategy to support the application of necessary security updates to areas that may impact the fulfilment of the functional, performance, and timing requirements of the system.

Whilst the MCP_Resource_Usage objectives naturally fall into SOI#2, it should be noted that the management of critical configuration settings for the hardware, particularly when they may be subject to change should be considered during the planning stages of using MCPs. A *Hardware Configuration Change Strategy* artefact, or similar, would therefore be beneficial for inclusion within SOI#1.

SOI#2 questions

Note: These questions are included here for completeness, but may well be answered within an associated AMC 20-152A argument.

1. How are inadvertent alterations to critical configuration settings (such as enablement/disablement of the I/O and cores or changes to memory/core speeds) managed?

2. How are inadvertent changes to critical configuration settings of the MCP prevented/mitigated?
3. Is a safety net, on-device and/or external monitor employed? *Note: this monitor may be in the form of CEH, such as a Complex Programmable Logic Device (CPLD) or Field-Programmable Gate Array (FPGA).*

Note: The objective might also include the allocation of certain inputs and outputs to specific cores, e.g. Intel VT-x changes for memory allocation whilst running.

4.5. 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

Discussion

This objective is, in part, achieved via the completion of MCP_Planning_1 and MCP_Planning_2 in conjunction with a justified design decision. Whilst verification techniques may have been explored as part of the MCP_Planning_1 and MCP_Planning_2 activities, they are not explicitly identified. Thus, for MCP_Resource_Usage_3 there is a need to ensure that a suitable and robust verification methodology is in place to either mitigate and/or eliminate interference channels. To reiterate from the associated note for this objective: interference may be *"caused by the use of shared memory, shared cache, an interconnect, or the use of any other shared resources, including shared peripherals"*; these should have been fully identified within the SOI#1 activities. The number of potential interference paths should not be underestimated [57–59].

A further associated note with this objective states:

> *"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".*

Although the applicant may have identified interference channels that do not impact the software in the final configuration, such channels should be treated with caution and justification provided due to the complexities of modern silicon devices. For example, there may be functionality within the silicon that is not documented [60] but has the potential to cause interference, and as such robust verification of the final configuration of the device will be required.

Evidence should be provided to demonstrate that sufficient control measures can be enacted through the methodology proposed. For example: what kind of cache policy is being applied and why? This could consider functionality such as cache disabling, cache locking, cache flushing, or cache partitioning. See US Army matrix [53] for further considerations. Equally, publications exist [61] which highlight techniques to reduce interference paths.

Addressing a further note (c) included against this AMC 20-193 objective, *"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"*. This note could be satisfied by the production of a suitable artefact such as a *life-cycle strategy for interference analysis* document, which was proposed in MCP_Planning_2.1.

There is a final self-explanatory note associated against this objective, summarised as: *for an IDAL C system (where safety analysis does not require robust partitioning), and if the applicant has chosen not to conduct an interference analysis, a justification for the decision should be provided. In this case the applicant should adhere to objective MCP_Software_1 (note c), as this will impact the manner in which software verification may be conducted.*

SOI#2 questions

1. Is the system IDAL C, and if so, has the applicant chosen not to conduct interference analysis, and how is this justified?
2. If the system is IDAL C (or higher) and the applicant is conducting interference analysis, then how is confidence obtained that the interference analysis is sufficiently complete?
3. Is a systematic or risk-based approach used to demonstrate completeness of the analysis?
4. How has the applicant identified the interference channels that could permit interference to affect the software applications hosted on the MCP cores?
5. How has the applicant assessed that the chosen means of interference mitigation cannot affect the software applications hosted on the MCP cores?

4.6. 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

Discussion

In interpreting this objective, the authors understanding is that the objective covers three areas for consideration. First, all available resources of the MCP, and of its interconnect (in the final configuration for the system design) have been identified. This configuration is established using the detail provided in MCP_Planning_2.1 and MCP_Resource_Usage_3.

Second, the allocation of the resources of the MCP to the software applications hosted on the MCP has been achieved, and is sufficiently documented. In support of this, the software architecture should be clearly identified, and the partitioning allocation across cores detailed for the proposed design. This should include supporting evidence that shows the rationale for the utilisation across cores remains within the specified bounds. Again, this is likely to employ evidence from prior interference analysis performed (such as that used for compliance demonstration of the planning (sub-)objectives). Note that fulfilment against this aspect of the objective may require further activities to be undertaken.

Whilst this objective is not required for IDAL C, it is recommended that the applicant considers (as good practice) the above aspects of this objective.

Third, and finally, verification evidence is required to illustrate that the demands for the resources of the MCP and of the interconnect do not exceed the available resources. Such verification evidence is required to ensure that: when all the hosted software is executing (on the target processor), its execution holds for all Worst-Case Execution Times (WCETs) scenarios. Definition of the worst-case scenarios in support of this objective, and how these are determined therefore needs to be shown. The worst-case scenarios would typically consist of maximum utilisation of the MCP system (or a part thereof), including normal and abnormal operation; but should not be the only situations considered. In considering WCET, the complexities of modern MCP should not be ignored; that is, features of the hardware make identifying a deterministic WCET significantly harder [62]. Therefore, the applicant could consider the probabilistic element to any WCETs calculations made, this would naturally lead to obtaining Worst-Case Probable Execution Times (WCPETs).

The authors also recommend that stress testing approaches [63] should be considered at this juncture to support later verification activities. Additionally, it may be prudent to consider early identification of any software component(s) or set(s) of requirements for which interference is precluded by design in support of MCP_Software_1.

If the system proposes an IMA approach then have the robust resource and timing constraints, as a consequence of IMA adoption, been satisfied.

SOI#2-3 questions

Note: As SOI#2 may be conducted prior to full completion of the design stage, some of these questions may not be fully addressed, and as such require re-review at SOI#3.

1. Have all of the available resources of the MCP, and of its interconnect, been identified within the final configuration?
2. Is the final configuration consistent with the detail articulated within MCP_Planning_2.1 and MCP_Resource_Usage_3? If there are any deviations, how have they been recorded and justified?
3. Does the proposed solution make use of IMA? If so, how does the intended final configuration meet the IMA constraints?
4. Are there any software requirements or components that do not impact the interference paths, and is there suitable rationale to support the claim?
5. Is the software architecture clearly identified and justified and is the partitioning allocation across cores detailed for the proposed design?
6. Has the allocation of the resources of the MCP to the software applications hosted on the MCP been fully defined and documented?
7. What evidence details and justifies the partitioning argument for the proposed design, and how does it align to any hypervisor/RTOS constraints?
8. What evidence demonstrates that the interference analysis has been performed to support the proposed design? *Note: this may be captured within a 'living document', such as a 'life-cycle strategy for interference analysis' (as proposed by the authors in MCP_Planning_2.1 in support of suitable design life-cycle interference analysis); if so, has this document been updated? Alternatively this may be captured in other suitable artefacts, generated specifically for this stage of involvement.*
9. Whilst a full verification artefact is not necessarily expected (at SOI#2), what evidence has been generated to support initial confidence that the demands for the resources of the MCP and of the interconnect do not exceed the available resources, when all hosted software is executing on the target processor, and will hold for anticipated WCET scenarios?

Note: whilst the majority of this objective may have been considered under SOI#2 for design aspects, SOI#3 should consider this objective for the Verification & Validation (VV) aspects as well, specifically the following:

10. What evidence demonstrates that the demands for the resources of the MCP and of the interconnect do not exceed the available resources, when all hosted software is executing on the target processor, and holds for the WCET scenarios?
11. What evidence demonstrates that sufficient, suitable, and justifiable scenarios have been defined for normal, abnormal, and WCET?
12. Has test evidence been generated to ensure that the scenarios have been satisfied? If stress testing is performed how has this been documented?

The applicability and completeness of response to these questions, at a given SOI activity, will ultimately be determined on a system-by-system basis, and at the discretion of the assessor.

4.7. 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

Discussion

The satisfaction of this objective covers the majority of the verification evidence that is likely to be provided in support of an argument toward AMC 20-193.

The reader may note (from the AMC) that, depending on the approach taken by the applicant for robust partitioning, the applicant should consider either Section 4.7.1 or Section 4.7.2. Specifically, Section 4.7.1 should only be considered where there is a robust partitioning solution for both spatial **and** temporal properties employed; as would be the case with an IMA approach. For non-robust partitioning solutions the fulfilment of this objective should follow Section 4.7.2.

4.7.1. MCP_Software_1: MCP platforms with robust partitioning

Discussion

This objective is felt to be self-explanatory: for systems employing robust temporal and spatial partitioning, then software application may be verified separately on the MCP, and the WCETs similarly separately assessed. Indeed, as per the AMC guidance note (d): *'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.*

SOI#3 questions

1. What evidence supports the claim that the MCP platform provides both robust spatial partitioning and robust temporal partitioning?
2. Does the robust partitioning conform to a strategy, such as that proposed in MCP_Planning_1.6?
3. Does the temporal resource partitioning support the WCETs identified for all software applications on the target hardware?
4. How has the applicant verified that all the software components hosted by the MCP comply with the applicable software guidance? *Note: this verification should include WCET results.*

4.7.2. MCP_Software_1: All other MCP platforms

Discussion

There may be scenarios where components of the software intended to be run on an MCP may not have any impact on, or utilise, interference channels. Whilst this is unlikely for the majority of software targeted for execution on an MCP, as per MCP_Resource_Usage_4, such software will still require verification activities in-line with the applicable software guidance.

Current MCP devices have the potential to provide spatial partitioning at the hardware level, for instance Intel's VT-x [64]. However, there may be arguments made where hardware capabilities for spatial partitioning are not leveraged (these should have been identified as part of MCP_Planning_2). Where spatial partitioning is not leveraged via certifiable hardware functionality, verification evidence must be provided, perhaps against a suitable SOI#1 artefact (e.g. a robust spatial partitioning strategy).

With respect to robust temporal partitioning, whilst there may be options through some software hypervisors and RTOS combinations, these may have the potential to significantly limit the utilisation of MCP devices. However, from the authors' understanding, robust temporal partitioning is far less likely to be achievable via currently available hardware features alone.

For the most part, illustration of sufficient temporal partitioning (such that interference paths are suitably mitigated) will need to be illustrated through processor profiling,[24] introduction of sufficient safety margins (as has previously been enacted for Single-Core Processors (SCPs) [65]), and extensive on target testing of the full, final software configuration. As per MCP_Resource_Usage_4, this extensive on target testing must consider normal and abnormal testing of each software component in conjunction with the rest of the software system. Target testing should also consider stress-testing techniques as well (outside of expected boundaries of operation). There has been significant work in both academia and industry to support the creation of demonic software components (e.g. [12,47]) to aid in this type of testing.

Fundamental to this objective is AMC 20-193's Note b, which states:

> *"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".*

From the above note it is expected that supportive evidence against this objective will be directed toward MCP robustness testing, and not that of individual software components. All the interfaces between the hosted software and the hardware of the MCP should be included in the (robustness) testing. Furthermore, as is identified within the AMC's associated note for this objective, Note e, there is the potential that *"interference may occur between tasks of a single component, when the tasks execute on different cores"*. The applicant should specifically consider the impact on interference paths for such cases.

Finally, a caveat, as introduced by Note c of this objective: if the MCP hardware and all of the software applications hosted on the MCP have a highest IDAL of IDAL C, and the applicant has not conducted an interference analysis (as per MCP_Resource_Usage_3 Note d), the hosted software components cannot be verified separately, and WCETs cannot be determined separately. In this case, it is necessary to perform testing *"on the target MCP with all software components executing in the intended final configuration, including robustness testing of the interfaces of the MCP"*. In the authors' opinion, this scenario is not desirable, and thus we encourage the fulfilment of MCP_Resource_Usage_3 for IDAL C as a means to ease compliance demonstration.

SOI#3 questions

1. Has the applicant defined the scheduling approach and provided sufficient rationale? *Note: for MCP platforms with robust partitioning, this may have been defined in MCP_Planning_1.6 [66].*
2. How has the applicant verified that all the software components hosted by the MCP comply with the applicable software guidance? *(Applicable software guidance: AC 20-115 [31], or equivalent, and any project-specific guidance).*
3. Has the applicant identified any software component(s) or set(s) of requirements for which the interference identified in the interference analysis is mitigated or is precluded by design? If so, how has this been verified?
4. How has the applicant verified that software components or sets of software requirements for which interference is not avoided or mitigated? *Note: these should be tested on the target MCP with all software components executing in the intended final configuration, including robustness testing of the all the interfaces of the MCP.*
5. How has the applicant ensured 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?
6. Have any additional interference paths been identified as part of this activity (for example derived in response to software architecture decisions) that were not previously identified, and have they been suitably captured and mitigated?
7. If tasks from a single component are allocated to different cores, has interference analysis been conducted to analyse and mitigate any interference issues between these tasks?
8. What evidence demonstrates that all interfaces between the hosted software and the hardware of the MCP have been included within the testing?

4.8. 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

Discussion

Definitions for data and control coupling are available within Annex B of DO-178C [1], given as:

> *"Data coupling - The dependence of a software component on data not exclusively under the control of that software component Control coupling - The manner or degree by which one software component influenced the execution of another software component".*

These are generally verified through a combination of: review and analysis of software architecture, review and analysis of source code, and requirements based testing confirmed by structural coverage analysis. The applicability of data and control coupling is equally applicable when implementing software on an MCP, and even more so across cores where temporal variability (brought about by interference) has the potential to impact these further.

Thus, not only must data and control coupling be considered for each core, but also across cores as the potential to introduce scenarios such as 'deadlock' and 'livelock' increase. Furthermore, there are potentials for stale, locked, and unsynchronised data scenarios that must also be considered and mitigated. As such, an overall approach or strategy to mitigating the ill-effects of data and control coupling may be warranted and could be defined within the planning stages.

It should be recognised that data and control coupling is non-trivial [67] when looking to provide sufficient verification evidence, and as such a *sympathetic* note is raised within the AMC:

> *"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".*

SOI#1 questions

1. Has a strategy been developed that considers data and control coupling which is supportive of the proposed architecture? *Note: this should include individual core and cross-core coupling.*

SOI#2 questions

1. Has the applicant identified data and control coupling between all the individual software components hosted on the same core of the MCP?
2. Has the applicant identified data and control coupling between all the individual software components hosted on different cores of the MCP?

SOI#3 questions

1. Has the applicant verified that the data and control coupling between all the individual software components has been exercised during software requirement-based testing?
2. In data and control coupling verification, has the applicant included exercising the interfaces between software components via shared memory and any mechanisms to control the access to shared memory?

[24] Processor profiling is discussed as part of the potential strategy for fulfilment of MCP_Planning_1.1.

3. Has the applicant verified that the data and control coupling in accordance with its strategy (e.g. with respect to the intended design and/or requirements definition)?
4. If verification is not completely possible at the software level, has evidence been provided of system-level tests that provide evidence in support of the objective?
5. In the context of data and control coupling, has the applicant performed an interference analysis for tasks of a single component that are executed on different cores?

4.9. 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

Discussion

The response to this objective is ultimately determined by the hardware and resultant software system chosen, hence the applicant has a significant amount of scope in which to respond. *Note: the term 'Safety net' is mentioned in this objective, it is defined within the AMC and is quoted within this paper in MCP_Planning_2.4*. Dependent on the final software configuration and capability of the hardware, the use of safety nets, which should be based on an assessment of the overall system, may be employed on:

- a separate area of the hardware device (such as Intel's Safety Island [68]),[25] or
- an external AEH device (utilising Watchdog Timers (WDTs), heartbeats, I/O monitoring etc.)

A complete solution to support error handling may involve using multiple safety nets (as shown above). This will ultimately entail suitable identification of failures and there detection, along with recovery plans/methods be employed.

The main aim of this objective is to provide measures where interference mitigations cannot be fully overcome, or where there may be shortfalls within the proposed solution. The shortfalls may not be failures of the system, but as a consequence of intentional features that cannot otherwise be overcome. For example, identification and management of thermal throttling of processor frequencies can be used to protect the device, however, this can adversely impact hard real-time requirements.

The considerations and proposed/implemented mitigations will be built up throughout the design life-cycle, and thus will likely be driven out of initial investigations under MCP_Planning_1, MCP_Planning_2, and continue through the subsequent SOI stages and associated objectives.

SOI#1-4 questions

1. What safety assessment has been performed to identify failures that may occur within the MCP and the effects on the overall system?
2. What fail-safe mechanisms have been identified that contain the effects of any failures within the equipment in which the MCP is installed?
3. What fail-safe mechanisms have been identified that contain the effects of any previously unmitigated features within the equipment in which the MCP is installed?
4. Has the applicant identified a strategy to allow the system and the MCP to fail in a safe manner?
5. Has a 'safety net' strategy been identified and how has the MCP system been implemented in accordance with this strategy?
6. How has the implementation of the 'safety net' strategy been verified?

The following considerations should be taken into account when satisfying this objective:

- This objective does not apply below IDAL B.
- The means to detect and handle failures in a fail-safe manner may include a 'safety net' independent from the MCP.
- The development of a possible safety net strategy should include interactions with stakeholders, such as system safety, software, hardware, and equipment engineers.
- The placement of safety nets should be supported by system safety analysis, which systematically looks at the system to identify where to place them.
- Safety nets may include any mechanism that monitors other processes, such as Built-In-Test (BIT), health monitoring systems, WDT, and parity checks.

4.10. MCP_Accomplishment_Summary_1: In addition to providing the information requested by the applicable software and AEH guidance, the applicant has provided documentation that summarises how they have met each of the objectives of the AMC

Discussion

This objective has similarities with the production of a Software Accomplishment Summary (SAS)/Hardware Accomplishment Summary (HAS) (as per DO-178C/DO-254), in that its fulfilment illustrates how the objectives have been achieved (at a high-level) and directs the reader to the location of the detailed evidence supporting the overall argument against the guidance/standard.

It may be beneficial for the applicant to include the MCP's specific plans and accomplishments within documents, such as the Plan for Software Aspects of Certification (PSAC), Plan for Hardware Aspects of Certification (PHAC), SAS, and HAS. If the applicant decides to have a dedicated set of MCP documentation, it may also be beneficial to follow the equivalent structure of this software and AEH guidance, and introduce a *'Plan for Multi-core Aspects of Certification'* and *'Multi-core Accomplishment Summary'* documents.

5. Evaluation and wider considerations

In developing the Job Aid questions contained within this paper, the authors acknowledge that the questions are linked to our interpretations of the AMC objectives and an associated mindset developed in response to in-depth group discussions. Whilst it is clearly not feasible to completely convey this mindset, the authors have attempted to articulate this within the discussions in the previous section. Ultimately, any implementation of guidance/standards will involve an element of interpretation of objectives to both develop and assess a system. Therefore, there is a strong reliance on the knowledge and experience of the individuals involved in such assessments.

5.1. Alignment of AMC 20-193 to SOI reviews

The authors' initial work [69] presented a tabular representation identifying the 4 SOI stages, against which the 10 objectives of AMC 20-193 were applicable. As a result of the further work undertaken, as presented in this paper, this position has been scrutinised and the updated position reflected in Table 1.

This mapping helped to align the objectives to stages and to structure the activities for AMC 20-193 review. The mapping also had the benefit of highlighting the iterative nature of the objectives through the SOI process, and will be discussed further in the next section.

[25] A separate core of the same device may also be employed as part of a wider safety net strategy.

Table 1
AMC 20-193 objectives mapped to SOI reviews.

	SOI#1	SOI#2	SOI#3	SOI#4
MCP_Planning_1	●			◗
MCP_Planning_2	●			◗
MCP_Resource_Usage_1		●	●	◗
MCP_Resource_Usage_2		●		◗
MCP_Resource_Usage_3		●		◗
MCP_Resource_Usage_4		●	●	◗
MCP_Software_1			●	◗
MCP_Software_2	●	●	●	◗
MCP_Error_Handling_1	●	●	●	●
MCP_Accomplishment_Summary_1				●

Key: ● = main objective effort; ◗ = confirmation effort.

5.2. Wider considerations

Whilst the following is not explicitly required by the AMC, these considerations may be classed as good engineering practice.

1. Whilst not always explicitly recognised within the aviation industry, there is a wider acknowledgement within the safety-critical world of the challenges brought about by the use of MCP devices. For instance, [70], Cerrozala et al. recognise that there is also a need to consider the *reliability* of the devices. As a result of the reduction in overall SWaP and the subsequent increased competition for shared resources, the reliability of the devices become paramount.
   The reduction in technology node sizes introduces further reliability challenges that should be considered: in the selection of the processor family, in the mitigation strategies, and in the 'safety nets' employed. These reductions in technology node sizes results in higher susceptibility to harsher environments [70], and indeed as has been witnessed in data centres, in the form of silent data corruptions [71] that are (in part) a consequence of these reduced node sizes. In [72], Lofwenmark and Nadjm-Tehrani identify the impact of fault injection techniques on the WCET, and highlight where areas of concern exist with respect to hardware integration and timing predictability.
   Whilst AMC 20-193 does not explicitly consider reliability of devices, this is a system-level consideration that needs to be addressed; particularly when attempting re-hosting activities for extant platforms.
2. Further consideration should be given towards AMC 20-152A COTS-4 with regard to the applicant's modification of Microcode; this is discussed further in Section 5.3.
3. The authors note that there is no explicit reference made to the processes underpinning the safety activities in support of generating the AMC 20-193 objectives and associated artefacts. That is, internal reviews/Quality Assurance (QA) processes, Configuration Management (CM) etc. are not highlighted. It is implicitly expected that these activities (detailed within DO-178C & DO-254) would be employed through the relationships to safety as illustrated in Fig. 1.
4. Whilst it is not the intent of this paper to delve into the detail, security is an important consideration with any solution, with safety informing security (and vice versa). However, again the AMC does not explicitly call out to any security considerations. Guidelines for security considerations do exist (DO-326A [73] and DO-355 [74]), however these have a system-level focus and may not consider the MCP-level. It would be remiss to not mention the security challenges of modern electronic devices, which are also applicable to AEH, such as side channel attacks. With the increasing connectivity, and complexity of avionics systems, security is becoming ever more important, and would, in the authors' opinion, warrant several papers to do this topic justice. This aside, it is beneficial to consider both safety and security in conjunction with other properties of a system, which are encapsulated under the term '*dependability*' [75]. Under the heading of dependability faults, errors and failures can be considered from both perspectives, along with measures of protection and mitigation.
5. Finally, an MCP implementation does not necessarily result in performance improvements, there are evidenced cases [76] of significantly degraded performance due to an MCP implementation, even when cache partitioning techniques are employed.

5.3. Insights from other MCP guidance

AMC 20-193 is the recognised means of compliance for MCP (as discussed in Section 3), however, within the defence domain, another means of compliance exists (AA 22-01 [29]). The authors note at this point that the level of granularity between the AMC and the AA is different, with the AA being more explicit, and what could be argued as more restrictive. From previous Defence Science and Technology Laboratory (Dstl) internal work, it was identified that 4 criteria within AA 22-01 can be considered advantageous[26] in addition to the objectives of AMC 20-193. These are discussed as follows:

- *"15.2.7.1 Susceptibility of the MCP to Single Event Upsets (SEUs) is identified. Effects of SEU occurrences on the MCP are detectable and mitigated by Software Processing Architecture (SPA) design"* Acknowledgement of this within a guideline is seen as key when considering modern devices. Although SEU effects are considered within system assessments, they are only *implicitly* considered within current civil guidelines.[27] This is also discussed within item (1) of Section 5.2.
- *"15.3.1.3 For cores supporting Safety Critical Functions (SCFs) , the lowest level cache is dedicated in its physical entirety to its respective core. Additionally, higher levels of cache are dedicated to cores either in their physical entirety or through use of a partitioning approach that allocates portions of cache to individual cores. Cache partitioning approaches do not unacceptably degrade SCF performance"* Dedicated lowest-level cache for each core is not explicitly identified as a requirement within AMC 20-193. There is potential intent as several of the objectives relate to reducing, controlling, or mitigating interference paths (such as MCP_Planning_2.1 and MCP_Resource_Usage_3). The explicit nature of this requirement ensures removal of one, if not the, most impacting interference channel.
- *"15.3.1.13 Microcode update mechanisms are identified and a control plan established that prohibits unauthorized updates by controlling and authorizing updates only after thorough evaluation and test. Microcode update procedures are repeatable, consistent, and include methods for verifying the correctness of the installed update. Updates do not have a detrimental impact on SCF processing"* Recognising that the intent of AMC 20-193 is to be enacted in conjunction with AMC 20-152A, an element of this criterion is addressed under objective COTS-4:

  > *If the microcode is not qualified by the device manufacturer or if it is modified by the applicant, the applicant should ensure that a means of compliance for this microcode integrated within the COTS device is proposed by the appropriate process, and is commensurate with the usage of the COTS device.*

  However, it is felt by the authors that, given the increased focus on security, that the COTS-4 objective may not go far enough. By contrast, AA 15.3.1.13 considers microcode not just from platform inception but that there may be a requirement for it to be updated through-life and managed accordingly. In doing so, additional activities in support of authoring qualified microcode must be

[26] Noting that AA 22-01 still has a draft status.
[27] AMC 20-152A provides an implicit link to EASA CM AS-004 [77].

considered. Such an update would also likely impact the resource utilisation of some, if not all, the software components (as we previously noted in reference to [55]); as identified within this AA criteria.

- *"15.5.3.7 Safety Supporting Software Elements (SSSEs) do not implement redundant functionality across cores/Virtual Machines (VMs) within the same physical MCP as a replacement for, or means to eliminate, physical redundancy"* This final differentiator between AA 22-01 and AMC 20-193 is perhaps the most important to raise. Whilst it may implicitly be considered within a system-level design decision, to ensure safety is preserved in the presence of hardware failure, redundant safety-critical software components must not reside on the same physical devices. As such, it is highly recommended that the intent of this criteria is upheld wherever practicable. In fact, the authors consider that this could be taken further, and that dissimilar hardware be used to avoid common mode failure [78].

5.4. Critique of approach

The methodology used, as set out in Section 1.1, provides a structured approach that uses SME expertise to provide an agreed understanding of the AMC objectives. However, it is recognised that the validity of this approach is limited by the experience and expertise of the SME participating in this study.

In creating a set of considerations, underpinned by current and credible material, a sound foundation for the development of a MCP Job Aid has been achieved. When undertaking this work it became apparent that there was a necessity to draw, not just from the formal literature (academic), but from grey literature (whitepapers, blog posts etc.) as well. This was necessary because, although there is an abundance of research discussing interference paths within MCP and their potential impact for safety critical systems, there is very little in the literature that readily supports understanding and answering the objectives as laid out in the AMC.

Thus, the authors have used caution and expert judgement when it was required to use non-academic sources, such that they support the output of this study. Furthermore, the intent of the paper has been to provide a harmonised interpretation of the AMC to support both the applicant and assessor.

Finally, due to the background of the SMEs in this study the outputs can be considered as having a Defence bias, when interpreting a civil standard. However, it is worth noting that defence regulation is often considered 'as civil as possible, as military as necessary'. As such, it is considered that the potential impact of this bias is minimal. In addition, the choice to undertake this work using SMEs from more than one nation provides more of a consensus view than that of a single nation.

Concerning the questions **Q1** and **Q2** posed in Section 1.1. **Q1** has been summarised by Table 1, which shows how the AMC objectives should be addressed within the various stages of the SOI reviews. As is evident from Table 1, whilst a structure can be applied to align to the development lifecycle it is not simplistic. There is a need to support more objectives earlier in the lifecycle than might be originally anticipated from an initial reading of the AMC. In fact, there is a continuing 'building' of artefacts and revisiting of objectives throughout the SOI engagements.[28]

In answering **Q2**, we would be remiss if we did not reflect on the potential impact of re-introducing Job Aids where others have been withdrawn, as highlighted in Section 2. As is identified throughout the paper, the use of MCPs for aviation is non-trivial, introducing complex challenges that span across multiple specialisms. Specifically, within the aviation domain, the use of MCPs are still considered to be a significantly novel technology. As the complexity increases with the use of MCPs the dangers associated with a 'checklist' solution are far outweighed, in the eyes of the authors, as it provides an effective method of structuring and assessing a complex system for certification. Furthermore, as has been highlighted, the questions introduced within Section 4 are by no means a complete set, that if answered, will lead to a successful certification argument against the AMC; each application of an MCP will require specific knowledge and a tailored approach. The work of this paper further indicates that there is no 'silver bullet' to certification of MCPs. It is considered that a suitable set of questions has been identified that provide context and a harmonised interpretation, sufficient to initiate and guide SOI interactions.

[28] As is expected, all artefacts are revisited in SOI#4 to some degree.

6. Conclusions, recommendations & next steps

This paper presents guidance that can be used *towards* forming a Job Aid, but not a finished Job Aid that can be used to assess MCPs. Furthermore, it does not provide a 'complete' question set to be used in a Job Aid. It provides supportive guidance to enable the key areas and salient points that should be considered when assessing and addressing the AMC's objectives.

The discussion and the questions in the paper are intended to provoke debate/dialogue within reviews, and to help to provide a harmonisation of the expectations of the objectives. Fundamentally, it is hoped that the paper will support a *common understanding* of the objectives' intent between regulators, assessors, and applicants.

When mapping AMC 20-193 objectives to the SOI reviews (Table 1) it was clear that the objectives are intrinsically interwoven. Some of the objectives (such as the management of critical configuration settings) require planning activities, but are implicitly associated with later SOIs. To efficiently demonstrate compliance to the objectives it is advised to identify these objectives and plan for them as best as possible, potentially even incorporating company strategy plans, like for hardware configuration changes. Whilst considered obvious, planning objectives are increasingly important: they lead the way towards compliance with the other objectives and help to resolve interpretation issues/ambiguities.

Although MCP certification may often be expected to be addressed by software, suitable and sufficient consideration must also be given to system and hardware aspects. The focus and expertise to deliver against AMC 20-193 requires the coordinated insights of multiple experts from different domains; importantly, cross-domain SQEP SMEs are needed.

In developing the paper, we have identified benefits to producing several strategy artefacts to aid certification compliance:

- *Life-Cycle Strategy for Interference Analysis*: Throughout the lifecycle there may be unexpected changes, in response to the development, to the intended final configuration. The ability to capture the approach to identification, analysis, assumptions raised, and the resultant interference paths, which may impact the system within a single artefact is considered advantageous.
- *Hardware Configuration Change Strategy*: It is not unreasonable to provide a method for managing changes to the hardware configuration, due to factors such as: the increasing complexities in hardware, the potential for increased microcode updates, and the potential need to perform through-life updates that may require the use of additional cores. How such changes will be assessed throughout the life of the system (not just the development) must be considered. Reasonably, any changes are unlikely to be driven from a safety perspective, but rather in response to security concerns or capability upgrades. The production of such an artefact would also be beneficial in supporting DO-326A and DO-355 security activities.
- *Data and Control Coupling Strategy*: An often misinterpreted concept for software is that of data and control coupling management and consequently there has been confusion about what is meant by the associated DO-178 objective [67]. The introduction of MCPs compounds the challenges, with considerations required for data and control of both the core and cross-core coupling.

- *Safety Net Strategy*: As has been discussed throughout this paper 'safety nets' may be required to mitigate hardware, software, or system-level shortfalls. However, in the authors' and others [79] experience, safety nets (if used incorrectly) have the potential to be detrimental to a safety argument rather than strengthen a position. Thus, it is fundamental to: specify and consolidate how safety nets are to be employed, when they are to be employed, and mitigate any detrimental impact.

Whilst we have chosen to use the word 'strategy' in the title of the above artefacts (thus emphasising their inclusion at the earliest stages of development) they should be considered as *living* documents, and should be updated as and when the need arises, in support of the overarching certification argument. In addition, the certification argument itself may need to be updated due to changes within the artefacts to reflect the two-way information interaction.

It is the intention of the authors to adopt the questions presented in this paper in their upcoming certification assessment work. Due to project-specific restrictions it is highly unlikely that detailed information may be shared, although it may be possible to share lessons learnt. It is the authors' hope that this paper provides sufficient details for practitioners to use this work, and thereby contributing to the use of MCPs for meeting airworthiness requirements in civil and military aviation.

Future endeavours could look to see how the questions generated within this work could be incorporated into proposed frameworks such as the Argumentation Pattern Notation [80], which provides a top down approach to structuring a certification argument [20]. For instance, the questions proposed here could well be introduced as templated constructs within such a model-based certification approach.

CRediT authorship contribution statement

James Sharp: Writing – original draft, Investigation, Conceptualization. **Mike Standish:** Writing – original draft, Investigation, Conceptualization. **Jaspal Sagoo:** Writing – review & editing, Investigation, Conceptualization. **Edwin van de Sluis:** Writing – review & editing, Investigation, Conceptualization.

Disclaimers and copyright

This paper is an overview of Dstl (UK MOD), QinetiQ, and Royal Netherlands Aerospace Centre (Royal NLR) research, which has been funded via separate projects. The paper is released for informational purposes only. The contents of this paper should not be interpreted as representing the views of Dstl, QinetiQ or Royal NLR, nor should it be assumed that they reflect any current or future UK or Dutch Government policy. The information contained in this paper cannot supersede any statutory or contractual requirements or liabilities and is offered without prejudice or commitment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is supported by several organisations which culminated in this paper, James Sharp and Mike Standish: UK Ministry of Defence, Jaspal Sagoo: QinetiQ Fellowship Scheme / UK Ministry of Defence DAT, and Edwin van de Sluis: Netherlands Ministry of Defence. Special thanks is warranted to Paul Nicholls for instigating this work and bringing us together as a team. Thanks to Rob Ashmore for his valuable insights from real-world experience. The authors would also like to thank Ian Hodgson from Rapita Systems & Tim Loveless from Lynx Solutions (for their interpretations on MCP_Planning_1.3, in particular note (b) and the intent behind IMA within the AMC). Their industry insights were particularly useful, and have aided the authors in framing the associated discussion. Finally, thanks to the reviewers who provided valuable feedback which improved the paper.

Data availability

No data was used for the research described in the article.

References

[1] RTCA, DO-178C. Software Considerations in Airborne Systems and Equipment Certification, RTCA, 2012.
[2] RTCA, DO-254. Design Assurance Guidance for Airborne Electronic Hardware, RTCA, 2000.
[3] D. Ypsilanti, OECD Observer, Volume 1985 Issue 1, OECD, 1985, p. 36, URL https://www.oecd-ilibrary.org/content/publication/observer-v1985-1-en.
[4] Grand View Research, Active electronic components market size, share and trends analysis report by product type (semiconductor devices, vacuum tubes, display devices), by end-user (consumer electronics, automotive), by region, and segment forecasts, 2023 - 2030, 2022, URL https://www.grandviewresearch.com/industry-analysis/active-electronic-components-market.
[5] IBM, Power 4: The first multi-core, 1GHz processor, 2023, WWW, URL https://www.ibm.com/ibm/history/ibm100/us/en/icons/power4/.
[6] Aerospace Tech Review, The evolving use of multicore processors in avionics, 2023, WWW, URL https://aerospacetechreview.com/the-evolving-use-of-multicore-processors-in-avionics/.
[7] R. Davis, D. Griffin, I. Bate, A framework for multi-core schedulability analysis accounting for resource stress and sensitivity, Real-Time Syst. 58 (2022) 456–508, http://dx.doi.org/10.1007/s11241-022-09377-8.
[8] B. Andersson, D. de Niz, Predictable use of multicore in the army and beyond, 2022, WWW, URL https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=884196.
[9] D. Iorga, T. Sorensen, J. Wickerson, A.F. Donaldson, Slow and steady: Measuring and tuning multi-core interference, in: 2020 IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS, 2020, http://dx.doi.org/10.1109/RTAS48715.2020.000-6.
[10] J. Nowotsch, M. Paulitsch, Leveraging multi-core computing architectures in avionics, in: 2012 Ninth European Dependable Computing Conference, 2012, pp. 132–143.
[11] J. Freitag, S. Uhrig, Quality of service for integrated modular avionics (IMA) on multicore processors using a safety net architecture, in: ERTS 2018, 2018.
[12] nHansa, Solutions for certification support, 2023, WWW, URL http://www.effective-automation.com/.
[13] S. Srinivasan, R. Kegley, M. Gerhardt, R. Hilliard, J. Preston, C. Granger, S. Drager, M. Anderson, R. Rosa, A. Charsagua, R. Ha, N. Srinivasan, Empirical bounds of multicore cache interference for real-time schedulability analysis, in: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference, DASC, 2019, http://dx.doi.org/10.1109/DASC43569.2019.9081787.
[14] X. Jean, D. Faura, M. Gatti, L. Pautet, T. Robert, Ensuring robust partitioning in multicore platforms for IMA systems, in: 2012 IEEE/AIAA 31st Digital Avionics Systems Conference, DASC, 2012.
[15] X. Jean, L. Mutuel, V. Brindejonc, Assurance methods for COTS multi-cores in avionics, in: 2016 IEEE/AIAA 35th Digital Avionics Systems Conference, DASC, 2016.
[16] Rapita & SYSGO, Developing DO-178c and ED-12c-certifiable multicore software, 2023, White Paper, URL https://www.rapitasystems.com/downloads/developing-do-178c-and-ed-12c-certifiable-multicore-software.
[17] EASA, AMC 20-193. Use of Multi-Core Processors, EASA, 2022, URL https://www.easa.europa.eu/sites/default/files/dfu/annex_i_to_ed_decision_2022-001-r_amc_20-193_use_of_multi-core_processors_mcps.pdf.
[18] CAST, CAST-32A. Multi-Core Processors, CAST, 2016, URL https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast-32a.pdf.

[19] A. Senechal, A(M)C 20-152A development assurance of airborne electronic hardware, in: Certification Together International Conference, 2023.

[20] A.F. Pires, F. Boniol, K. Delmas, B. Lesage, A.M. Gonzalez, C. Pagetti, T. Polacsek, Combining argumentation-based and model-based approaches for the certification of multi-core architectures: The PHYLOG methodology, in: 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference, DASC, 2024, pp. 1–10.

[21] EASA, AMC 20-152A. Development Assurance for Airborne Electronic Hardware (AEH), EASA, 2020, URL https://www.easa.europa.eu/sites/default/files/dfu/amc-20_amendment_20.pdf.

[22] FAA, Conducting Software Reviews Prior to Certification. Job Aid, FAA, 2004.

[23] FAA, Conducting Airborne Electronic Hardware Reviews. Job Aid, FAA, 2008.

[24] EASA, CM-SWCEH-001. Development Assurance of Airborne Electronic Hardware, EASA, 2018, URL https://www.easa.europa.eu/en/document-library/product-certification-consultations/easa-cm-swceh-001.

[25] G. Ferreira, C. Kastner, J. Sunshine, S. Apel, W. Scherlis, Design dimensions for software certification: A grounded analysis, 2019, arXiv:1905.09760.

[26] J.A. Jimenez, J.A.M. Merodio, L.F. Sanz, Checklists for compliance to DO-178C and DO-278A standards, Comput. Stand. Interfaces 52 (2017) http://dx.doi.org/10.1016/j.csi.2017.01.006.

[27] V. Hilderman, The Aviation Development Ecosystem: Applying DO-178C, ARP4754A, DO-254, and Related Guidelines, Afuzion Incorporated, 2021.

[28] MAA, MAA02: Military Aviation Authority Master Glossary. Issue 11, Tech. rep., MAA, 2022.

[29] USAF, Airworthiness Advisory (AA)-22-01. Multicore Processor Environments Supporting Safety Critical Functions, USAF, 2022.

[30] US Army, Software Airworthiness Qualification Requirements for Multi-Core Processors. S3I-SWAW-MCP-DOC-001, Tech. rep., US Army, 2021.

[31] FAA, AC 20-115D. Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( ), FAA, 2017, URL https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-115D.pdf.

[32] FAA, AC 20-152A. Development Assurance for Airborne Electrionic Hardware, FAA, 2022, URL https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-152A.pdf.

[33] CAST, CAST-32. Multi-Core Processors, CAST, 2014.

[34] SAE, ARP-4761. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, SAE, 2010.

[35] SAE, ARP-4754A. Guidelines For Development Of Civil Aircraft and Systems, SAE, 2006.

[36] MAA, Military air system certification of multi-core processors, 2023, WWW, URL https://www.gov.uk/government/news/military-air-system-certification-of-multi-core-processors.

[37] EASA, AMC 20-170. Integrated Modular Avionics (IMA), EASA, 2020, URL https://www.easa.europa.eu/sites/default/files/dfu/amc-20_amendment_20.pdf.

[38] M. Herlihy, N. Shavit, V. Luchangco, M. Spear, The Art of Multiprocessor Programming, Morgan Kaufmann, 2021.

[39] Power.org, Standard for Embedded Power Architecture Platform Requirements, Version 1.1, IEEE-ISTO, 2011.

[40] J. Treibig, G. Hager, G. Wellein, Multi-core architectures: Complexities of performance prediction and the impact of cache topology, 2009, CoRR abs/0910.4865, arXiv:0910.4865, URL http://arxiv.org/abs/0910.4865.

[41] NXP, Running AMP, SMP or BMP mode for multicore embedded systems, 2012, White Paper, URL https://www.nxp.com/docs/en/brochure/PWRARBYNDBITSRAS.pdf.

[42] M. Strobl, Virtualization for Reliable Embedded Systems (Master's thesis), 2013.

[43] T. Chen, L.T.X. Phan, SafeMC: A system for the design and evaluation of mode-change protocols, in: 2018 IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS, 2018, pp. 105–116, http://dx.doi.org/10.1109/RTAS.2018.00021.

[44] H. Omar, H. Dogan, B. Kahne, O. Khan, Multicore resource isolation for deterministic, resilient and secure concurrent execution of safety-critical applications, IEEE Comput. Archit. Lett. 17 (2) (2018) 230–234, http://dx.doi.org/10.1109/LCA.2018.2874216.

[45] A. Mazouz, S.-A.-A. Touati, D. Barthou, Performance evaluation and analysis of thread pinning strategies on multi-core platforms: Case study of SPEC OMP applications on intel architectures, in: 2011 International Conference on High Performance Computing and Simulation, 2011, pp. 273–279, http://dx.doi.org/10.1109/HPCSim.2011.5999834.

[46] RTCA, DO-297. Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations, RTCA, 2005.

[47] S.H. VanderLeest, C. Evripidou, An approach to verification of interference concerns for multi-core systems (CAST-32A), SAE Int. J. Adv. Curr. Pr. Mobil. 2 (2020-01-0016) (2020) 1174–1181.

[48] S. West, S. Nanz, B. Meyer, Demonic testing of concurrent programs, in: T. Aoki, K. Taguchi (Eds.), Formal Methods and Software Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 478–493.

[49] RTCA, DO-330. Software Tool Qualification Considerations, RTCA, 2011.

[50] Tim Loveless, What is a separation kernel?, 2020, WWW, URL https://www.lynx.com/embedded-systems-learning-center/what-is-a-separation-kernel.

[51] Y. Dong, X. Yang, J. Li, G. Liao, K. Tian, H. Guan, High performance network virtualization with SR-IOV, J. Parallel Distrib. Comput. 72 (11) (2012) 1471–1480, http://dx.doi.org/10.1016/j.jpdc.2012.01.020, Communication Architectures for Scalable Systems, URL https://www.sciencedirect.com/science/article/pii/S0743731512000329.

[52] G. Gracioli, A. Alhammad, R. Mancuso, A.A. Frohlich, R. Pellizzoni, A survey on cache management mechanisms for real-time embedded systems, ACM Comput. Surv. 48 (2) (2015) http://dx.doi.org/10.1145/2830555.

[53] US Army, MCP interference matrix, 2023, WWW, URL https://www.youtube.com/watch?v=QI34HBJ99kA.

[54] Intel, What is intel management engine?, 2023, WWW, URL https://www.intel.com/content/www/us/en/support/articles/000008927/software/chipset-software.html.

[55] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, Y. Yarom, Spectre attacks: Exploiting speculative execution, in: 2019 IEEE Symposium on Security and Privacy, SP, 2019, pp. 1–19, http://dx.doi.org/10.1109/SP.2019.00002.

[56] M. Cinque, D. Cotroneo, L. De Simone, S. Rosiello, Virtualizing mixed-criticality systems: A survey on industrial trends and issues, Future Gener. Comput. Syst. 129 (2022) 315–330, http://dx.doi.org/10.1016/j.future.2021.12.002, URL https://www.sciencedirect.com/science/article/pii/S0167739X21004787.

[57] F. Boniol, F. Thurieau, A mixed model-based and instrumentation-based approach for the certification of multi-core avionics computers, in: Certification Together International Conference, 2023.

[58] B. Lesage, C. Pagetti, PHYLOG2 dealing with certification issues of hybrid platforms (multi-core with hardware accelerators), in: Certification Together International Conference, 2023.

[59] F. Crestey, Use of multicore processors in certified avionics: Inherent specificities and resulting challenges, in: Certification Together International Conference, 2023.

[60] Y. Yang, P. Qiu, C. Wang, Y. Jin, Q. Gao, X. Li, D. Wang, G. Qu, Exploration and exploitation of hidden PMU events, in: 2023 IEEE/ACM International Conference on Computer Aided Design, ICCAD, 2023, pp. 1–9, http://dx.doi.org/10.1109/ICCAD57390.2023.10323695.

[61] T. Lugo, S. Lozano, J. Fernandez, J. Carretero, A survey of techniques for reducing interference in real-time applications on multicore platforms, IEEE Access 10 (2022) 21853–21882, http://dx.doi.org/10.1109/ACCESS.2022.3151891.

[62] I. Bate, R. Reutemann, Worst-case execution time analysis for dynamic branch predictors, in: Proceedings. 16th Euromicro Conference on Real-Time Systems, 2004, ECRTS 2004, 2004, pp. 215–222, http://dx.doi.org/10.1109/EMRTS.2004.1311023.

[63] M. Hadley, M. Standish, A practical assurance approach for multi-cores (MCs) within safety-critical software applications, in: SSS20 Safety-Critical Systems Symposium, 2020.

[64] Intel, Intel virtualization technologies, 2023, White Paper, URL https://www.intel.com/content/dam/www/central-libraries/us/en/documents/intel-virtualization-technologies-white-paper.pdf.

[65] J. Gustafsson, A. Ermedahl, Experiences from applying WCET analysis in industrial settings, in: 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, ISORC'07, 2007, pp. 382–392, http://dx.doi.org/10.1109/ISORC.2007.36.

[66] G. Gilliand, Safety-critical multi-core for avionics, in: SSS22 Safety-Critical Systems Symposium, 2022.

[67] L. Rierson, Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance, CRC Press, 2017.

[68] J. Abella, F.J. Cazorla, S. Alcaide, M. Paulitsch, Y. Peng, I.P. Gouveia, Envisioning a safety island to enable HPC devices in safety-critical domains, 2023, arXiv:2307.11940.

[69] J. Sharp, M. Standish, J. Sagoo, M. Kastelein, M. de Bruijn, E. van de Sluis, Interpreting AMC 20-193 objectives: Toward a job aid for compliance assessment and demonstration, in: Certification Together International Conference, 2023.

[70] J. Perez-Cerrolaza, R. Obermaisser, J. Abella, F. Cazorla, K. Gruttner, I. Agirre, H. Ahmadian, I. Allende, Multi-core devices for safety-critical systems: A survey, ACM Comput. Surv. 53 (2020) http://dx.doi.org/10.1145/3398665.

[71] G. Papadimitriou, D. Gizopoulos, H.D. Dixit, S. Sankar, Silent data corruptions: The stealthy saboteurs of digital integrity, in: 2023 IEEE 29th International Symposium on on-Line Testing and Robust System Design, IOLTS, 2023, pp. 1–7, http://dx.doi.org/10.1109/IOLTS59296.2023.10224870.

[72] A. Lofwenmark, S. Nadjm-Tehrani, Fault and timing analysis in critical multi-core systems: A survey with an avionics perspective, J. Syst. Archit. 87 (2018) 1–11, http://dx.doi.org/10.1016/j.sysarc.2018.04.001, URL https://www.sciencedirect.com/science/article/pii/S1383762117304903.

[73] RTCA, DO-326A. Airworthiness Security Process Specification, RTCA, 2014.

[74] RTCA, DO-355. Information Security Guidance for Continuing Airworthiness, RTCA, 2014.

[75] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Dependable Secur. Comput. 1 (1) (2004) 11–33, http://dx.doi.org/10.1109/TDSC.2004.2.

[76] H. Yun, K. Valsan, Evaluating the isolation effect of cache partitioning on COTS multi-core processors, in: Proc. of the 6th Nordic Conf. on Human-Computer Interaction, in: 11th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications, 2015.
[77] EASA, Single Event Effects (SEE) Caused by Atmospheric Radiation. EASA CM No.: CM-AS-004 Issue 01, EASA, URL https://www.easa.europa.eu/sites/default/files/dfu/CM-AS-004Issue01.pdf.
[78] R. Ashmore, M. Hadley, J. Sharp, et al., Reducing the risk of a software common mode failure, Safety-Crit. Syst. EJournal 2 (2) (2023).
[79] J. Ganssle, Chapter 26 - building a great watchdog, in: J. Ganssle (Ed.), The Firmware Handbook, in: Embedded Technology, Newnes, Burlington, 2004, pp. 339–354, http://dx.doi.org/10.1016/B978-075067606-9/50033-3, URL https://www.sciencedirect.com/science/article/pii/B9780750676069500333.
[80] K. Delmas, C. Pagetti, T. Polacsek, Patterns for certification standards, in: Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Grenoble, France, June 8-12, 2020, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2020.

James Sharp Ph.D. focus looked at verification of VHDL using Formal Methods titled "Shared Variable Analyser For Hardware Descriptions", which attempted to improve the scalability of the application of Model Checking to enable meaningful test. Post Ph.D., James spent 5 years working on formal verification of high-criticality hardware within Defence prior to joining Dstl. Since joining Dstl James has become an accredited Independent Technical Evaluator (ITE), focussing on Software assurance for Air Platforms. In addition, James actively undertakes SOI assessments for MOD in association with UK Novel Complex Electronic Hardware component. His open publication list can be found here: https://orcid.org/0000-0003-2213-0925.

Mike Standish is a Principle Scientist in systems at the UK Defence Science and Technology Laboratory (Dstl). Mike has experience of all aspects of software and systems lifecycles, which has been gained from nearly 20 years within the defence sector. Mike provides software/CEH SME technical support to a number of MOD air platforms. Mike holds a B.Sc. in Software Engineering and a M.Sc. in Strategic Information Systems. Mike also holds an Engineering Doctorate (EngD) in Systems from the University of Bristol with a focus on adopting wider diverse evidence to mitigate shortfalls in software process-based safety assurance evidence. He is a Chartered Engineer (CEng) gained via the British Computer Society (BCS). His open publication list can be found here: https://orcid.org/0009-0009-2917-0883.

Jaspal Sagoo is a Principal Consultant with over 30 years' experience in providing assurance for systems based on software and programmable hardware that are used in avionics and automotive domains. At QinetiQ, Jaspal holds a fellowship, which recognises his expertise at a national/international level, and he performs reviews (for regulatory authorities) on suppliers showing compliance to standards such as DO-254 and DO-178C. Currently, Jaspal is visiting professor at Aston University, and he has held several academic positions that involved directing/managing/conducting research into real-time embedded systems. Jaspal is a reviewer for several international journals, the UK EPSRC, and he has publications in the assurance of dependable systems. Jaspal serves on several University Industrial Advisory Boards, IET panels for CEng/IEng and Academic Accreditation.

Edwin van de Sluis, is a Senior Software/System Engineer in the Flight Test & Certification Support department of the Royal Netherlands Aerospace Centre (NLR). Edwin graduated in 1988 at the State University of Utrecht, as a computer scientist. After a post-graduate course at the technical University of Eindhoven and fulfilling his military duties, he started in 1992 his career at NLR as an embedded software engineer. During this career he evolved into a software/system engineer specialising in the field of safety-critical systems and software. As such he has been involved in the development and verification of software up to DO-178B level A. Since December 2015 Edwin is a Compliance Verification Engineer (CVE) with The Netherlands MoD with respect to system and software certification aspects.