DEPARTMENT OF THE AIR FORCE
HEADQUARTERS AIR FORCE LIFE CYCLE MANAGEMENT CENTER
WRIGHT-PATTERSON AIR FORCE BASE OHIO

ADVISORY
AA–22–01
26 Oct 2022

AIRWORTHINESS ADVISORY

Multicore Processor Environments Supporting Safety Critical Functions

1. Purpose. This Airworthiness Advisory (AA) provides air system program offices with the information needed to understand, plan, and implement the additional Computer Systems and Software (CS&S) Airworthiness (AW) verification activities required when utilizing a Multicore Processor Environment (MCPE) to support the processing of a Safety Critical Function (SCF). This AA is applicable to all versions of MIL-HDBK-516[1].

2. Scope. This AA applies to all air systems owned, leased, operated, used, designed or modified by the United States Air Force (USAF), the Air National Guard, and USAF Reserve.

3. Cancellations. NONE.

4. Referenced Documents.

[1] MIL-HDBK-516C, *Airworthiness Certification Criteria*, 12 December 2014

[2] AWB-100, *Airworthiness Bulletin 100, Airworthiness Process Overview and Terminology,* 3 June 2021

5. Attachments.

[1] DRAFT Airworthiness Verification Standards to Address the Use of Multicore Processor Environments Supporting Safety Critical Functions, 9 August 2022

6. Background. In a digital computer, the Central Processing Unit (CPU) is the hardware component responsible for general purpose processing and execution of instructions (e.g., as defined in a software program). The key component within a CPU that performs this processing and execution is the processing core. USAF air systems have historically utilized digital computers containing Single Core Processors (SCPs); CPUs with a single processing core. To assess the Computer Systems and Software (CS&S) airworthiness of these types of air system architectures and designs, a set of verification criteria (see [1] MIL-HDBK-516C, Section 15) has been established and utilized for many years.

Recently, an industry shift amongst CPU, Single Board Computer (SBC), and similar vendors has occurred in response to changing market demands. These vendors have moved away from providing traditional SCP offerings, instead shifting to Multicore Processors (MCPs). MCPs are

[1] This AA makes specific reference to MIL-HDBK-516C in several places and while not always specifically mentioned, it is also applicable to programs utilizing other versions of MIL-HDBK-516

CPUs that contain more than one processing core, which allows for multiple instructions to be processed and executed in parallel. In light of this and the diminishing availability of SCPs, air system vendors have followed suit. The utilization of MCPEs (collective term used to refer to an MCP, its settings and features, interfacing resources, and software control structure [e.g., operating system, hypervisor]) is becoming increasingly prominent within air system CS&S architectures and designs, to include those supporting the processing of SCFs (ref. SCF definition in [2] AWB-100).

7. Discussion. While the use of MCPEs can provide for increased flexibility and processing throughput, their incorporation into CS&S architectures and designs, particularly in support of SCFs, introduces a new set of technical challenges. The hardware designs of MCPs (e.g., common system resources [e.g., memory, input/output (I/O)] amongst competing processing cores) can lead to non-deterministic operation and increased execution times, resulting in latent and/or erroneous SCF functionality and performance. In addition, multiple MCPE architectures (e.g. symmetric, asymmetric, bound) exist that can be used in varying combinations and configurations, some implementations being better suited for supporting SCFs than others. The incorporation of MCPEs also introduces new failure modes and the potential for increased susceptibilities (e.g., to Single Event Upsets [SEUs]).

Failure to properly account for these challenges and to design and verify/validate accordingly can result in assessments of increased risk due to non-compliances with current MIL-HDBK-516C, Section 15, airworthiness verification criteria. Utilization of MCPEs on air systems requires program offices to supplement existing MIL-HDBK-516C criteria with additional or tailored standards and methods of compliance in order to substantiate airworthiness for the air system.

8. Recommendations. Air system program offices should coordinate the use of MCPEs supporting SCF processing with AFLCMC/EZAS CS&S Subject Matter Experts (SMEs) early in the air system's development lifecycle. These SMEs have developed and utilized a draft set of standards and methods of compliance (see Attachment 1) to supplement existing MIL-HDBK-516C, Section 15 language when MCPEs supporting SCF processing are being incorporated into an air system's CS&S architecture and design. Program offices should incorporate these additional standards and methods of compliance, designed to balance the flexibility of MCPEs with the airworthiness verification expectations of CS&S, into the air system's Certification Basis (CB).

9. Point Of Contact. The Office of Primary Responsibility (OPR) for this AA is the Integrated Avionics, Computer Systems and Software Branch (AFLCMC/EZAS). Comments, suggestions, or questions on this AA should be directed to: Mr. Christopher Jackson,

christopher.jackson.1@us.af.mil; or Mr. Christopher Pfeifer, christopher.pfeifer@us.af.mil. General AW questions should be directed to the USAF Airworthiness Office, USAF.Airworthiness.Office@us.af.mil

JANNING-LASK.JACQUELINE.SUZANNE.1230137079

Digitally signed by JANNING-LASK.JACQUELINE.SUZANNE.1230137079
Date: 2022.11.08 14:59:12 -05'00'

JACQUELINE S. JANNING-LASK, SES, USAF
Director, Engineering and Technical Management/Services (AFLCMC/EN-EZ)
USAF Technical Airworthiness Authority

ATTACHMENT 1

DRAFT Airworthiness Verification Standards to Address the Use of Multicore Processor Environments Supporting Safety Critical Functions, 9 August 2022

BACKGROUND:
In 2016, AFLCMC/EZAS Airworthiness (AW) Subject Matter Experts (SMEs) began an effort to investigate the need for modifications to MIL-HDBK-516C, Section 15 to address airworthiness considerations when utilizing Multicore Processor Environments (MCPEs). While the results of this investigation did not identify the need for any additional criterion, it was determined that a number of criteria would require additional Standard and Method of Compliance (MoC) language to address the airworthiness considerations necessary when performing verification of MCPEs supporting Safety Critical Functions (SCFs). Those criteria are identified in the table below.

INSTRUCTIONS:
For each of the identified Section 15 criterion, a draft set of Standard and MoC language is provided. Programs should add this language to an air system's CB when an MCPE is being used to support SCF processing. **Note that when adding this language into a CB, the language should be added as an addition to existing MIL-HDBK-516C, Section 15 Standard and MoC language for a given criterion; it is not intended as a replacement.**

To assist in understanding, an explanation for each column in the table below follows:

- "Criterion" column - A Section 15 criterion that AFLCMC/EZAS SMEs have determined is impacted by the use of MCPEs, for which additional Standard and MoC language is needed to account for additional airworthiness considerations.

- "Additional Standard Language" column - The additional Standard language that should be added to the criterion in the "Criterion" column. Each additional Standard has an associated MoC in the "Additional Method of Compliance Language" column.

- "Additional Method of Compliance Language" column - The additional MoC language that should be added to the criterion in the "Criterion" column. Each additional MoC has an associated Standard in the "Additional Standard Language" column.

The specific Standard and MoC language that should be added to an air system's CB depends on the Section 15 criteria that are applicable to that air system, in addition to the configuration of the MCPE being implemented. For MCPEs that are configured to only utilize a single core, a subset of the additional Standards and MoCs may apply. Any questions on the additional Standards and MoCs, applicability, or other concerns should be directed to Mr. Christopher Jackson, email at christopher.jackson.1@us.af.mil; or Mr. Christopher Pfeifer, email at christopher.pfeifer@us.af.mil. For air systems utilizing versions of MIL-HDBK-516 other than 'C', contact the above for guidance.

NOTE: In some instances, the additional Standards and MoCs have applicability to SCPs as well. This will be addressed in a future update to MIL-HDBK-516.

DRAFT, 9 August 2022

Criterion	Additional Standard Language	Additional Method of Compliance Language
15.1.4	For each Safety Critical Function (SCF) identify: all MCPs utilized, the designated Computer System Integrity Level (CSIL) for each MCP, processor cores utilized, and chip related resources (e.g., interconnects, I/O management, controllers, memory, buses) supporting the operation and performance of the SCF.	The Safety Critical Function Thread Analysis (SCFTA) identifies for each SCF supported by an MCP the CSIL of the MCP(s) and all testing, simulation, and analysis needed to adequately verify and validate (V&V) the ability of the cores and chip resources to support safe, deterministic, fault tolerant SCF operation.
15.2.3	The integration methodology adequately accounts for the safety aspects of utilizing the MCP environment.	Verify through inspection, analysis, and test that the integration methodology provides for adequate coverage of all safety, functional, and performance aspects of MCP development, integration, and test.
15.2.5	Simulators, models, and tools used to emulate or otherwise support development or test of MCP processes and environments have been adequately verified and validated for their intended use, to include the generation of supporting airworthiness compliance evidence. Tools utilized to represent the performance of MCP environments are qualified to requirements and validated to show that their performance is equitable to the actual flight configuration.	Analysis and test verify that simulations, models, and tools used to model the MCP environment contain the required fidelity to accurately model the configuration, behavior and interactions of the MCP environment.
15.2.7	Susceptibility of the MCP to Single Event Upsets (SEUs) is identified. Effects of SEU occurrences on the MCP are detectable and mitigated by System Processing Architecture (SPA) design.	Analysis of the MCP identifies susceptibility of SEU and consequence of occurrence. Analysis and test verify that SEU events are detectable and safely addressed by mitigations employed.
15.3.1	The processor family, model, and architectural and performance details (clock rate, total number of cores, etc.) of each MCP have been documented including known processor errata and deficiencies. Any known uses of the MCP in safety critical applications have been documented to include known issues with its use in the given applications. Documented MCP errata, deficiencies, and issues have no safety impacts to SCF operation or have been mitigated and shown to be safe.	Analysis verifies that all MCP errata, deficiencies, and issues have been analyzed for safety impacts and mitigations established as needed to ensure safe SCF operation.
15.3.1	The cores and other resources that are hardware enabled and hardware disabled in the MCP are identified. Methods used for hardware disabling are identified. Hardware enabled cores and other resources that are deactivated by MCP configuration settings maintain the configured state throughout system operation. Mechanisms are in place to detect and accommodate inadvertent state changes to support continued safe SCF operations.	Inspection of MCP configuration documentation and settings verifies that cores that are hardware enabled and hardware disabled have been identified and confirms how cores will be hardware enabled/hardware disabled. Analysis and test verify that the mechanism to enable and disable cores maintains the configuration as set. Analysis and test verify that inadvertent changes to state are detected and safely accommodated to support continued safe SCF operation.
15.3.1	For cores supporting 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.	Inspection of MCP design and configuration documentation verifies that lowest level cache is dedicated in its entirety to its respective core. Inspection verifies that higher levels of cache are dedicated to individual cores either in their physical entirety or through a partitioning approach. Analysis and test verify that the cache architecture and any partitioning scheme(s) utilized support safe SCF operations.
15.3.1	Data coherency is implemented and maintains the integrity (value and timing) of data associated with safety critical processing across the MCP environment.	Analysis and test verifies that the data coherency mechanization maintains the integrity (value and timing) of data used by SCFs.
15.3.1	Unused cores within an MCP are hardware disabled or deactivated. Core activation/deactivation mechanisms ensure deactivated cores remain deactivated and activated cores remain active. The MCP environment performs periodic monitoring during operation to ensure core activation/deactivation scheme does not change. Fault handling techniques ensure continued safe operation in the presence of changes in activation state (e.g., deactivated core becomes activated).	Inspection of MCP configuration settings confirm how cores will be activated/deactivated, and that unused cores are disabled or deactivated. Analysis and test verify that the mechanism to activate and deactivate the cores maintains the configuration as set. Testing verifies that the MCP environment periodically monitors the state of core activation/deactivation and detects any changes in the activation state. Testing verifies that fault-handling techniques provide continued safe operation for failures that change the activation state.
15.3.1	The failure modes of all cores, chip resources, and components of an MCP, whether hardware enabled, hardware disabled, activated, or deactivated, have been identified, are understood, and mitigations are in place for ensuring continued safe SCF operation in the event of their occurrence.	Failure Modes and Effects Analysis/Failure Modes, Effects and Criticality Analysis (FMEA/FMECA) documents failure modes of the MCP and identifies their potential impact to SCFs. Analysis, testing, and simulation of chip level failures verifies that failure mitigations are in place to support continued safe SCF operation per safety and fault tolerance requirements. Analysis, testing, and simulation of chip level failures verifies that failures of cores, chip resources, and components not supporting SCFs do not impact SCF operations, or that their effects have been mitigated to support continued safe SCF operation.
15.3.1	The MCP HW contribution to the processing of SCFs is shown to be safe, deterministic, and in compliance with latency and performance requirements.	Analysis and test verifies that the hardware provides deterministic operation, and meets latency and performance requirements for SCF operation.
15.3.1	Interface management approaches, between cores and external to the MCP, are identified for each core to include rates, prioritization, routing, error management, and interference resolution.	Analysis verifies that the interface management approaches utilized by the MCP (between cores and external to MCP) have been identified and provide adequate rates, priorities, routing, error management, and interference resolution for SCF operation. Testing, including Failure Modes and Effects Testing (FMET), verifies that the functional operation and performance of the interface management approaches meet requirements and are adequate for SCF processing.
15.3.1	Interference channels and areas of possible resource contention have been identified with their associated mitigations necessary to support safe SCF operation. Processing conditions that can increase the likelihood of interference or resource contention are identified.	Analysis of the MCP hardware identifies all interference channels and areas of resource contention, processing conditions that can contribute to interference or resource contention, and associated mitigations. Testing verifies that the MCP responds as expected under conditions that can produce interference or resource contention and that mitigations employed are adequate to ensure safe SCF operation. The testing is performed over a comprehensive set of expected operating conditions including the known worst case operating and timing conditions that can result in interference or resource contention. The testing covers all known interference channels, areas of possible resource contention, and associated mitigations.

15.3.1	Failure modes of the MCP have been identified along with their detectability. Critical failure modes (modes that can impact SCF operation) have been identified and are detectable. Non-critical failure modes are shown to not impact SCF operation. Detectable failure modes are shown to be detected and safely managed by the system. The system provides a safe response to MCP failure modes (single or combination) in accordance with fault tolerance requirements. All failures that are undetectable during power-on/off, initialization, checkout, and flight operations (PICFO) have been identified and have a means to be detected during ground maintenance operations.	Failure analysis (i.e., FMEA, FMECA) verifies the MCP failure modes have been identified, the detectability of each failure mode, and whether each failure mode is a critical failure mode. Analysis and testing, including FMET, verify the detection of failure modes and the system response supports fault tolerance requirements, including ensuring that non-critical failure modes have no impact on SCF operation. Inspection of support documentation verifies that ground maintenance actions exist to detect failure modes that cannot be detected during PICFO. Testing verifies that identified ground maintenance actions detect failure modes that cannot be detected during PICFO.
15.3.1	A safe approach is used for managing the features and settings of each core and resource in an MCP. All configurable features and settings of the MCP are identified and the flight configuration of the MCP settings documented. The deactivation of cores and resources within an MCP through the use of configuration settings and features does not impact safe SCF operation. Mechanization of MCP settings ensures settings remain persistent and have an Improbable likelihood of inadvertent change during operation. Fault conditions that can result in changes to the MCP configuration settings are detectable and safely accommodated. All dynamically modifiable performance features of the MCP (e.g., dynamic modification of clock rates), to include all features that can impact deterministic processing, are identified and how each will be configured to eliminate or minimize dynamic processing changes in order to support SCF processing. All power saving features of the MCP are disabled for non-ground maintenance operations; ground maintenance operations may utilize power saving features provided the operations are not impacting the processing of an SCF (e.g., an interlock preventing unsafe emission). Features of the MCP that have the potential to cause the processing to be non-deterministic or create undetermined interference channels or resource contention are disabled.	Inspection of MCP design and configuration documentation verify that the following are identified: configurable features, dynamically modifiable performance features, and the defined configuration of the settings for all features. Analysis and test verify that the deactivation of cores and resources within an MCP through configuration settings and features does not impact safe SCF operation. Analysis and test verifies that the MCP settings mechanization scheme maintains the settings per the Standard and that the settings utilized support deterministic, stable SCF processing and disable features that can influence deterministic operation (e.g., power saving features), cause interference channels, or resource contention. Analysis and testing shows that unintended changes in configuration settings are detected and safely accommodated.
15.3.1	The MCP is designed to fully support an Asymmetric Multi-Processing (AMP) or Bound Multi-Processing (BMP) scheme for cores processing a Safety Supporting Software Element (SSSE). The MCP does not permit (by design or configuration settings) the dynamic allocation of a process/VM to another core processing an SSSE. Processing schemes used on cores not processing an SSSE do not have any nominal or failure mode impacts to SSSE processing.	Analysis and testing verify that the MCP supports an AMP or BMP scheme for cores processing an SSSE and that other schemes utilized for cores not processing an SSSE have no safety impact on SSSE processing. The analysis also shows that the MCP will not dynamically allocate a process/VM to another core processing an SSSE.
15.3.1	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.	Inspection of Configuration Management (CM) documentation and microcode update procedures verify that control plans and update procedures are in place for safely handling microcode updates, including fully evaluating the impact to SCF processing. Testing verifies that microcode updates can be performed repeatedly, consistently, and verify the installation is correct. Testing verifies that the microcode update does not have a detrimental impact on SCF processing.
15.3.1	The MCP hardware interrupt configuration/design is identified. MCP hardware interrupts are serviced in a safe and deterministic manner that does not impact other non-interrupted cores/VMs that are operating in the MCP environment. Interrupt routing and servicing schemes utilized provide for deterministic, stable SCF operations. Shared interrupts are not utilized for SCF operations (exception: a full MCP reset interrupt). Interrupts supporting a non-SCF process do not impact SCF processing.	Inspection of MCP design and/or interrupt configuration documentation verifies that the configuration approach for hardware interrupts has been identified. Analysis and test verify that the routing and servicing scheme facilitates deterministic, stable SCF operations; does not impact non-interrupted cores/VMs; and that non-safety critical interrupt handling does not impact safety critical processing.
15.3.1	The interconnect design, routing, and arbitration scheme utilized to interface all cores and resources within the MCP provide for deterministic, stable SCF operation. The design and routing utilized do not introduce unacceptable latency or cause unacceptable jitter. Arbitration of requests, messages, etc. supports deterministic SCF operation compliant with latency and performance requirements. Failures in non-safety critical processes do not impact the ability of the interconnect, its routing, or arbitration to support safe SCF operations.	Inspection of the MCP design documentation verifies that the interconnect design, routing, and arbitration schemes have been defined. Analysis and test verify that the design and routing employed do not introduce unacceptable latency, or unacceptable jitter, into messages, requests, etc. supporting SCF operations. Analysis and test also verify that the arbitration scheme utilized prioritizes those messages, requests, etc. supporting safety critical processing to ensure compliance with latency and performance requirements and that failures with non-safety critical processes do not impact the ability of the interconnect to support safe SCF operations.
15.4.1	The software development processes must account for developing SSSE software to operate within the MCP environment.	Add the following to the list in the MoC of 15.4.1: a. Consider usage within the MCP environment
15.4.3	The CM process accounts for managing the configuration of MCP environments including modifiable MCP parameters and microcode.	No additional MoC needed
15.5.1	The software architecture implemented across the cores of an MCP is documented.	No additional MoC needed
15.5.1	The software architecture accounts for the unique attributes associated with operation in the MCP environment. The software architectural design identifies: all cross-application and cross-core dependencies, within and external to, the MCP; processing scheme (e.g., AMP, Symmetric Multi-Processing (SMP)) planned for each core; SSSE allocation to cores and VMs; MCP resource allocation to software architectural components; resource control and management (e.g., bus control software); hypervisor implementations; virtual machines utilized; partitioning schemes utilized and the Operating System (OS)/executive capability to support multicore operation to comprehensively show the software architecture's interaction and operation with MCP hardware.	Inspection of software and MCP design documentation verifies that the software architecture accounts for operating within the MCP environment. Analysis verifies that the software architectural design accounts for the items identified in the Standard and that the software architecture and hardware interaction and operation is fully understood.
15.5.1	Any given instance of an application running in a core/VM processing an SSSE is assigned to only one core/VM and the assignment remains static during operation. The dynamic allocation of a non-SSSE application to a core/VM not processing an SSSE does not impact safe SCF operation.	Analysis of the software architecture within the MCP environment verifies that the assignments of application instances to a core/VM processing an SSSE are appropriate and remain static during operation. Analysis and test verify that the dynamic allocation of a non-SSSE application to a core/VM not processing an SSSE does not impact safe SCF operation.
15.5.1	MCP resources needed for SSSE operation are allocated and identified as shared or dedicated resources.	Analysis verifies that MCP resources needed by each SSSE are identified and designated as shared or dedicated.

15.5.1	Single core SSSEs that have been ported over for reuse on the MCP are identified and properly modified for use in the MCP environment.	Inspection of software process documentation verifies that a sound approach was used to port single core SSSEs to the MCP environment. Analysis of the SSSE identifies the attributes that are accounted for to port to the MCP environment, or that no modifications are needed. Testing verifies that the function and performance of the ported software meets functional and performance requirements.
15.5.2	The Standard identified in 15.5.2 is applicable per each core/Virtual Machine (VM) within an MCP supporting SCF processing. All instances of executive control coupling between cores/VMs are identified and shown to support safe SCF operation.	Inspection of software design documentation verifies the software executive structure for each core/VM is identified, and that all instances of executive control coupling between cores/VMs has been identified. Testing verifies Operating systems/executives are adequately verified and validated for use in the MCP environment. Analysis and testing verify that hypervisors are adequately verified and validated and shown to be safe and compatible with the chosen operating system(s). Analysis and testing under nominal and off-nominal conditions verify that all instances of executive control coupling between cores/VMs support safe SCF operation.
15.5.2	For each core/VM utilized within the MCP environment, the software control and execution structure of the hosted software elements are identified, to include: a. Processing rates for each process/thread b. Foreground vs background process structure and prioritizations c. Major and minor frame structures and scheduling d. Memory management approach for each process e. Synchronization techniques used to synchronize cores/VMs to other cores/VMs and from cores/VMs to redundant processes external to the MCP. The software control and execution structure are designed to meet system performance requirements and support safe SCF operation.	Inspection of software design documentation verifies that the software control and execution structure attributes identified in the Standard have been identified. Analysis and testing verifies that the software control and execution structure meets requirements and is safe.
15.5.2	The OS/Executive enforces an AMP or BMP scheme for the operation of all cores processing an SSSE.	Inspection of software architecture requirements and design verify that an AMP or BMP scheme is utilized for cores processing an SSSE. Analysis, demonstration, and test verify that the OS/Executive control enforces an AMP or BMP scheme for cores processing an SSSE.
15.5.2	The software control and execution structure accounts for interference channels and resource contention, and provides adequate mitigations for safe SCF operation.	Analysis verifies that interference channels and resource contention are identified for the MCP environment and their potential impact to SCF operations. Analysis and testing verify that the software control and execution structure provide adequate mitigations for every possible interference channel and resource contention identified that can impact SCF operations.
15.5.2	All software interfaces with MCP hardware are identified and support safe SCF operation.	Inspection of MCP hardware-software interface design documentation verifies that the software interfaces with the MCP hardware have been identified. Testing verifies that the interface design supports SCF processing requirements and there are no issues with the compatibility of the interface design.
15.5.3	The Standard identified in 15.5.3 is applicable per core/VM within the MCP.	No additional MoC needed.
15.5.3	The MCP SW contribution to the processing of SCFs is shown to be safe, deterministic, and in compliance with functional, timing, and performance requirements.	Analysis verifies that the MCP software design does not have unacceptable hazards; is designed to provide deterministic operation; and addresses allocated functional, timing, and performance requirements needed for safe SCF operation. Testing verifies deterministic operation, adequacy of hazard mitigations, and that requirements are met by the software design.
15.5.3	Data and control flow are identified for all internal (core/VM-to-core/VM) and external interfaces and cross-process dependencies. The methods used for data sharing and messaging are identified. All instances of data and resource sharing between (1) MCP processes and (2) all cores/VMs have been identified. Cores not processing an SSSE do not have write capability to shared safety critical data. All functional coupling between processes running in the MCP environment are identified and support safe SCF operation.	Inspection of software design documentation verifies that data sharing and messaging methods have been identified. Analysis verifies that data and control flow have been identified per the Standard, all instances of data and resource sharing between (1) MCP processes and (2) all cores/VMs are identified, and that cores not processing an SSSE do not have write capability to shared safety critical data. Functional coupling analysis verifies that all inter-process coupling (i.e., direct and indirect via data or control coupling) within the MCP environment has been identified and that safety critical processing does not have any coupling dependencies on non-safety critical data or control flow.
15.5.3	Data integrity and coherency mechanisms are incorporated for data and messaging management within the MCP environment. Data integrity and coherency mechanisms meet design requirements and are adequate to support safe SCF operation.	Inspection of software design documentation verifies the data integrity and coherency mechanisms are identified for data/messaging. Analysis and testing verify that the mechanisms used provide the integrity and coherency needed to meet design requirements and support safe SCF operation.
15.5.3	Software within the MCP environment that can contribute to the occurrence of, or mitigate the impacts of, interference channels and/or resource contention events (e.g., shared resources, deadlocks, race conditions) is identified. Mitigations are employed to address identified interference channels and resource contention to ensure safe SCF operation in the event of occurrence.	Analysis verifies that all software within the MCP environment that contributes to or mitigates occurrences of interference channels and/or resource contention have been identified. The analysis addresses all software within the MCP environment (e.g., application software and software related to: memory management schemes, processing frame structures and timing, data and control flow, and other internal and external dependencies). Analysis and test verify that the mitigations employed to address interference channels and resource contention support safe SCF operation.
15.5.3	Within the MCP environment, the methods for synchronizing safety critical processes and threads across cores/VMs (internal and external to the MCP) are documented and support safe SCF operation.	Inspection of software design documentation verifies that the synchronization approach for the MCP environment is documented. Analysis and test verify that the safety critical processes and threads are properly synchronized and meet SCF processing and fault tolerance requirements.
15.5.3	SSSEs do not implement redundant functionality across cores/VMs within the same physical MCP as a replacement for, or means to eliminate, physical redundancy.	Inspection of design documentation verifies that SSSEs do not implement redundant functionality across cores/VMs within an MCP as a replacement for, or means to eliminate, physical redundancy.
15.5.3	Timing requirements for processes have been established for each active core within the target MCP environment. The Worst Case Execution Time (WCET) has been identified for each process and meets latency and performance requirements. All interference channels; interfaces with other applications and cores; resource contentions; and signal, data, and timing dependencies have been accounted for. WCETs are acceptable for safe operation of supported SCFs, and account for any cumulative WCET conditions supporting an SCF.	Inspection of requirement documents verify that timing requirements for processes running in the MCP environment have been established. Analysis verifies that WCETs have been identified for each process in the target MCP environment, accounts for all applicable contributors, and identifies applicable cumulative WCET conditions that could impact SCF operation. The analysis assesses a process WCET with respect to all other processes running in the MCP environment, taking into account any impacts that the other processes may have on the WCET. Analysis and testing verifies that the WCETs (including cumulative conditions) support latency and performance requirements.
15.5.3	Software allocation and partitioning within each core/VM is shown to be safe. Partitioning implementations for each core/VM are identified and ensure that all partitioned software have protected time and data processing that cannot be interfered or interrupted by software executing in other partitions. The partitioning scheme ensures that all throughput and data resources for SSSE processing are fully protected. The CSIL of the software performing the partitioning scheme is equal to or greater in criticality to the highest level CSIL running within the partitioning scheme.	Inspection of design documentation verifies the partitioning approach that is utilized for each core/VM is identified and that CSIL levels are appropriately assigned to the software performing the partitioning scheme. Analysis and test verify that software operating (nominally or in a failed state) in other partitions do not have the ability to influence partitions running an SSSE. The analysis and test show that time and data processing of the SSSE are protected. Testing includes nominal operation testing and FMET to ensure failure conditions in other partitions do not propagate or effect functionality and/or performance of partitions running an SSSE.
15.5.3	The MCP environment initialization approach is identified. The approach correctly initializes the MCP environment upon startup or restart.	Inspection of design documentation verifies that the initialization approach is identified. Testing verifies that the MCP environment initializes correctly upon startup or restart.
15.5.3	All buses supporting SCF operation (i.e., safety critical buses) have bus messaging traffic software (i.e., bus controllers) running on cores/VMs that are designed to run an SSSE. Redundant bus control functionality for a safety critical bus is implemented across separate MCPs; a physical MCP does not contain more than one core/VM for controlling the same safety critical bus. MCP bus resources utilize a deterministic bus control mechanization for SSSE operations, while providing adequate performance to support safe SCF operation.	Inspection of design documentation verifies safety critical buses are identified, are controlled by cores/VMs that are designed to run an SSSE, and that a physical MCP does not contain more than one core/VM for controlling the same safety critical bus. Testing verifies that MCP bus resources utilized by an SSSE are deterministic and support safe SCF operation.

15.5.4	The Standard identified in 15.5.4 is applicable per each core/VM within an MCP supporting SCF processing.	No additional MoC needed
15.5.5	The Built-In-Test (BIT) implementation is compatible with the MCP environment and supports fault tolerance requirements.	No additional MoC needed.
15.5.6	The MCP environment is capable of detecting and accommodating digital system failures (including all unique MCP failure modes) to meet safety and fault tolerance requirements.	Analysis verifies that the MCP failure modes are identified. Analysis and testing (i.e., FMET) verify that the identified MCP failure modes are detected and accommodated to meet fault tolerance requirements.
15.5.6	The watch dog monitor designs and protection mechanisms for each process and core/VM are fully documented, prevent failure propagation within SCF operation, and do not induce safety hazards. The triggering of watch dog monitors or protection mechanisms by a core not processing an SSSE or by a non-safety critical process does not impact safe SCF operation.	Inspection of design documentation verifies that the monitoring and protection mechanisms have been documented. Analysis and test verify that the watch dog monitors and protection mechanisms prevent failure propagation, do not induce hazards, and ensure that triggering by a core not processing an SSSE or due to a non-safety critical process does not impact safe SCF operation.
15.5.6	Any detection of an MCP hardware failure, accommodates the failure with a graceful shutdown of the entire MCP environment (exception: if it can be assured that the failure detected is a hardware failure limited to a single core and the accommodation will not have any impact to SCF processing in other cores, then this particular failure mode may allow the MCP environment to continue processing while only gracefully shutting down the core that failed).	Analysis verifies failure modes that are clearly hardware failures of the MCP are identified and whether the failure can be isolated to drive an accommodation that shuts down one core (i.e., failure does not impact SCF operation) or the whole MCP. Simulation and/or testing verifies that hardware failure modes result in graceful shutdowns of MCP (or core when applicable).
15.5.7	The Standard identified in 15.5.7 is applicable per core for the implementation of restart or reset capability within the MCP environment.	No additional MoC needed
15.5.8	Techniques that can cause non-deterministic behavior within an MCP environment are prohibited, examples include the following: a. Dynamic SCF process/thread assignment across cores, b. MCP execution thread parallelization (e.g., Simultaneous Multi-Threading (SMT)) for SCF processing, c. VMs containing SCF processing that rely on hypervisors that require software emulation of individual core operations, d. Run-time MCP configuration changes once initialization is complete, e. Dynamic MCP clock rate modifications, f. MCP power saving modes (except for Ground Maintenance as noted in 15.3.1).	No additional MoC needed
15.5.9	For MCPs, throughput utilization of processing cores and system bus (used to connect processing cores to system resources) does not exceed 90% under worst case loading conditions. Throughput utilization assessments of processing cores should measure throughput for each minor frame that performs critical timing requirements for SCF processing.	Analysis and testing under worst case loading conditions involves accounting for the worst case conditions that all software running in the MCP environment could realistically encounter with all software running simultaneously. When testing is not possible to assess internal aspects of the MCP, analysis of those characteristics is an acceptable verification method.
15.6.1	The integration methodology adequately accounts for the safety aspects of utilizing the MCP environment. Examples of areas addressed include resource and process allocation to cores; cross-core application and process dependencies, integration, and test; interference channels and resource contention; performance of shared resources; configuration and design validation of MCP operation with all loaded software components; and core-level failure mode coverage.	Inspection of software development and software integration plans verify that proper planning for utilizing and verifying software running on MCP environments has been accomplished.
15.6.2	SSSEs running on MCPs are fully qualified with the MCP configured and running with the full flight configuration of software loaded on the MCP and running in as close to the worst case-processing configuration as possible that the system can realistically experience. Test environment simulations/stubs can be utilized to create the full flight configuration of loaded software if those simulations/stubs are qualified to requirements and validated to show that their performance is equitable to the actual flight configuration.	Inspection of SSSE qualification plans, procedures, and reports verify that the qualification was performed with a full flight configuration of software planned for the MCP environment. Analysis and testing verify that any simulations/stubs meet acceptable performance within the qualification environment. Analysis of test results verify that the simulations/stubs functioned as expected during the qualification process.
15.6.3	The software build process accounts for unique build aspects of MCP and verifies that the build result meets all safety objectives of the build that are required for the software to operate safely in the MCP environment.	Inspection of build process documentation verifies that the build process has properly accounted for unique MCP build requirements. Analysis and testing of the software build verifies that the build is correct and has incorporated all necessary safety attributes to meet required safety objectives.
15.6.5	The software load process accounts for unique loading aspects of the MCP environment and verifies that the load result meets all safety objectives that are required for the software to operate safely in the MCP environment.	Inspection of software load process documentation and loading instructions verifies that the load process has properly accounted for unique MCP loading requirements. Analysis and testing of the software load procedures validates that the load process is correct and has incorporated all necessary safety attributes to meet required safety objectives.