Position Paper

On

Minimal Multicore Avionics Certification Guidance

Lui Sha and Marco Caccamo
University of Illinois at Urbana-Champaign

Greg Shelton, Marc Nuessen, J. Perry Smith, David Miller and Richard Bradford
Rockwell Collins Inc.

Russell Kegley, Dennis Perlman and Jonathan Preston
Lockheed Martin Corporation

Joseph M. Wlad and Mathew Storr
WindRiver Systems

Dionisio de Niz, Sagar Chaki, Mark Klein and Bjorn A. Andersson
Software Engineering Institute
Carnegie Mellon University

Iain Bate and Alan Burns
University of York

Stuart Palin
BAE Electronic Systems, UK

Stanley Bak, Derek Kingston, Matthew Clark
Aerospace Systems Directorate, Air Force Research Lab

Taehoe Kim and Eunji Pak
ETRI, South Korea

August 4, 2016

Acknowledgment: This and earlier versions have been sent to Michael DeWalt for reviews.

Background: The inter-core interference in multicore chips is a major challenge in the development and certification of multicore avionics. Many of the interference topics have already been identified by existing documents such as FAA CAST-32[1] and a draft EASA CRI, published in 2014.

[1] https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-32.pdf

Objective: To identify *necessary requirements* on multicore avionics certification process that will maximize the freedom to develop innovative solutions without sacrificing quality and consistency of the certification process. A broad-based consensus on the necessary certification requirements will promote consistency in the certification of multicore avionics and is a step towards the development of a standardized guidance in the future.

Scope: This guidance assumes that meeting stringent timing requirements are needed. It focuses on the software aspects of multicore certification and implementation of multiple levels of criticality using software or hardware virtualization techniques. The guidance here does not address complex electronic hardware requirements that may involve certification to RTCA/DO-254. Nor does it address security and fault tolerance concerns. It is recognized that some aspects of software certification of multicore systems indeed require the hardware model of the multicore chip. The use of COTS products may require both supplier and integrator to collaborate to achieve guidance objectives. Finally, solution methods to meet the requirements are outside of the scope of this document.

1.0 Certification Requirement 1: Modular Certification of Core Group

A core group consists of one or more cores. It has its own set of reserved computing resources shared by cores, such as the number of cores, DRAM banks, memory bus bandwidth, size of last level shared cache, and shared I/O channel bandwidth. The configuration of core groups is application dependent.

Shared resources between core groups must be partitioned in such a way that the certification process can ensure that applications in each core group are certified independently. It is unacceptable to permit applications in one core group to invalidate the certification of applications in other groups[2]. This is especially important when core groups are used by applications with different levels of criticality.

Core group with one core: A core group with only one core **shall** be certified as if it were a single core chip. When more than one core is used, the worst case inter-core interference **must** be both bounded and accounted for in each task's worst case execution time (WCET) analysis and validation. This allows the reuse of the schedulability analysis and validation under DO-178 B/C. However, certain hardware architectures may not allow effective bounding of worst case inter-core interferences.

Core group with two or more cores: When a core group consists of two or more cores, the delay caused by the cache coherence protocol and by potential intercore interferences, the implications of using a memory consistency model weaker than sequential consistency, and the potential invalidation of real time synchronization protocols developed for single core chips **must** be analyzed. The correctness of proposed solutions of these parallel processing problems **must** be validated as part of the certification process.

Note that incorrect sharing of resources can create significant delay spikes even in single core chips, such as the well-known uncontrolled priority inversion problem. The incorrect use of shared cache can also induce significant delay spikes on single-core processors too. Fortunately, the avionics community has developed certifiable solutions for the resource sharing problem of single core chips. Multicore chips bring in new challenges. This

[2] Without the per group certification, the avionics certification has to wait until all the core groups applications are done. In addition, after the certification if any software is modified in any group, certified software in other groups may need to be recertified.

guidance adds additional and necessary requirements for the multi-core avionics certification process.

2.0 Certification Requirements 2: Memory Models, Partition Mechanisms and Multicore Parallel Processing

2.1 Validation of Memory Model and Partitioning Mechanisms

Commercial multicore processor designs are driven by the need for good average performance in general-purpose applications. They typically do not include provisions to isolate or protect all needed resources used by a given application. Processing interference is common between the cores often in the form of resource sharing among cores. These consequences are not usually seen in multiple processor designs and are not of keen interest to commercial developers except a few. For developers of safety-critical systems, the consequences can be significant and impact safety requirements. For example, timing delays of up to 600% caused by inter-core interference were documented in the Lockheed Martin Space Vehicle Integration Lab in a Freescale P4080 testbed[3].

Inter-core partitioning of shared resources requires more attention and information about the underlying hardware model and particular processor implementation than seen heretofore under DO-178B/C. The certification of multicore avionics **should** address the following:

- **Interference channels**: An undesirable multicore platform property. It refers to the interference between independent applications that reside in different cores. Interferences are the result of applications accessing resources that are concurrently shared by cores. Common sources of potential inter-core interference channels include the sharing of a DRAM bank between cores, the sharing of memory bus bandwidth, the sharing of last-level cache, the sharing of the I/O channels, DMA and the sharing of the on-chip network. However, this is not a complete list. Developers need to investigate the processor architecture and identify all the interference channels.

  DO-178 B/C was developed in the context of single core chips. Unless the inter-core interference across each channel is bounded and accounted for, applications in a core group may be difficult to certify independent of other core groups. Furthermore, reuse of DO-178 certification data for a core group of size one may be difficult or impossible.

- **Mitigations**: Are defined as implemented mechanisms to reduce or eliminate the impact of the interference channels on the hosted software so that the software can meet its performance and availability requirements. The design data available from the processor supplier **should** be combined with the applicant's interference channel analysis and measurements. The usage scenario **shall** be designed to enable the validation process to quantify the worst case interference channels' impact on the hosted software's performance and availability.

- **Memory model**: A multicore chip allows concurrent accesses to the memory system by different cores. Multicore CPUs also allow the use of a memory consistency model that is weaker than the sequential consistency model used by single-core chips. Since DO-178B/C was developed for single-core chips with a sequential consistency model, the

[3] Source: L. Sha, M. Caccamo, R. Mancuso, J.E. Kim, M.K. Yoon, R. Pellizzoni, H. Yun, R. Kegley, D. Perlman, G. Arundale and R. Bradford, Single Core Equivalent Technology for Hard Real-Time Computing on Multicore Processors, to appear in IEEE Computer.

memory model did not create an added certification burden. It is recommended that a sequential consistency model be reused until such time as more guidance becomes available for those using the multi-core consistency model. The memory model is discussed in more detail in Section 2.2, below.

- **Module interface enforcement**: When modules share cores, the level of interference defined in the module interface **should** be enforced in order to prevent a violation of an interference interface of one module from affecting another module, breaking the isolation requirement.
- **SafetyNet**: When more than one core is used, the inter-core interference channel's impact on worst case execution time (WCET) falls outside of the traditional single core WCET analysis and validation process. There is a general concern documented in CAST-32 and draft EASA guidance that given the complexity of multi-core processors' resource sharing logic, there could be residual interferences not fully bounded and accounted for in some corner cases. To account for this, a validated system level partitioning design monitor **should** be required to detect timing fault, so that application logics can handle the fault in a way that is similar to the handling of frame overrun in single-core processors.

2.2 Multicore Parallel Processing

Certain applications cannot fit into a single-core environment and be able to meet their performance requirements. The certification of applications using multi-core parallel processing **should** address further certification challenges including:

- **Interference channels**: When a core group has two or more cores, the potential interference channel effects discussed above must be bounded and accounted for during the certification.
- **Memory consistency models**: When multiple cores are used by an application, the sequential consistency model has significant impact on performance and typically weaker consistency models are used in general purpose applications. The certification of these weaker models (if used) is NOT part of traditional uniprocessor certification process. The development of a certification procedure to explicitly validate the memory model/behavior of a multicore chip **should** be required.
- **Real Time synchronization protocols**: The properties of traditional real time synchronization protocols such as the priority ceiling protocol supported by many RTOS will NOT hold when tasks/threads sharing the semaphores are running on different cores. There are multi-processor extensions of the single processor priority ceiling protocol that may be adopted for multicore processors. However, they are outside of the scope of DO178B/C certification process developed for single core chips.
- **Cache coherence protocols**: In a core group with N (N > 1) cores, the worst case delay caused by the implemented cache coherence protocol **should** be taken into account.
- **Protection and Isolation**: In a single core chip, an IMA architecture provides shared resource isolation and protection between applications in different IMA partitions. IMA is undefined when a core group has two or more cores. When there are

multiple applications, the resource isolation and protection mechanisms **should** be certified.

- **Task/thread migration**: In a single core chip tasks/threads cannot migrate. When more than one core is used migration between cores is possible. When migration is allowed, resource isolation and protection mechanisms should be certified. Prohibiting migration is an acceptable option.
- **Parallel processing interfaces must** be defined and enforced in order to preserve modular certification.
- **Inter Core-Group Communication:** Dependencies between core-groups could be created by inter Core-Group Communication. Such dependencies must be analyzed and taken into account during certification.

3.0 Certification Requirement 3: Acceptable Performance and Availability Impact by Interference Channels

On a multi-core processor several resources, such as DRAM memory, memory controller, last level cache, on-chip network, DMA and I/O channels are shared by cores. Each is a potential interference channel. Unless a shared resource is over-provisioned[4] or partitioned[5], the inter-core interference can be very large and hard to determine.

For all interference channels that have a potential impact on the performance and availability of hosted application software, the certification process **must** validate that 1) the interference channels have been individually identified and their potential impact on the performance and availability of hosted applications, after mitigation, has been quantified, and 2) the potential total impact of all interference channels on application software performance and availability, after all mitigations, is accounted for and validated in the schedulability analysis.

Quantifying the potential impact of interference channels on application software performance and availability, combined with validated partitioning (see above) is the foundation for modular certification. Bounding the impact from interference channels on performance and availability is also the foundation to compute if deadlines can be reliably met.

4.0 Industry Recommendation

Solution approaches are fundamentally predicated on the data provided by the chip manufacturer. Unfortunately, the behavior of a typical commercial processor as described in the user manual may be incomplete, contain errors or have ambiguities that allow for incompatible interpretations. These information and design gaps make certification of software on multicore devices difficult if not impossible. To address these challenges, an advisory committee sponsored by a professional organization, such as RTCA and EUROCAE, **should** establish:

a. Chip design data that is accurate and sufficient to allow the appropriate analysis to take place.

[4] A shared resource is said to be "over-provisioned" if the maximal delay resulted from using this resource is acceptably small and accounted for under maximal workload.

[5] A shared resource is said to be "partitioned" between cores, if each core can only use a limited percentage of the shared resource expressed as time budget X in interval Y, using certain hardware and/or software isolation mechanism.

b. A framework for identifying and compartmentalizing design elements such that interference channels are clearly identifiable by the applicant, and

c. Design guidelines to minimize the creation of new interference channels by the processor vendor.

d. A minimum set of acceptable design data and certification artifacts that demonstrate the requirements (including partitioning considerations) of DO-178B/C can be met if multi-core processors are used in safety-critical applications.