Alternative Multi-Core Processor Considerations for Aviation

Scott Gerhold  
Principal Electrical Engineer  
Rockwell Collins, Inc.  
Cedar Rapids, IA, USA

Mike Dunham  
Principal Systems Engineer  
Rockwell Collins, Inc.  
Cedar Rapids, IA, USA

Branden Sletteland  
Principal Software Engineer  
Rockwell Collins, Inc.  
Cedar Rapids, IA, USA

ABSTRACT

Power Architecture® processors have dominated aviation safety-critical processing since the late 1990s, when major processing vendors exited the MIL-qualified and/or aviation-certified markets. Since that time, four trends have emerged:

1. Military and commercial safety certification has become more rigorous
2. Server/desktop architectures have focused on performance at the expense of determinism
3. System-on-Chip (SoC) architectures are offered, with multiple processing cores (multicore) in a single package to increase performance over single-core processors
4. The industrial automation industry is increasing safety requirements for autonomous manufacturing, and the automotive industry is offering driver assistance, including autonomous operation, creating a large market for relatively low-power, high-integrity processing

Although the Power Architecture will remain a viable aviation processor technology for some time, new-to-our-industry processing products and architectures are poised to enter (or re-enter) the aviation market. Automotive and aviation markets require similar capabilities that make the avionics market attractive to processing vendors currently supplying the automotive market:

- Longer product availability lifetimes (5-15 years) than consumer/server-grade processors
- Low power draw
- Extended temperature operation
- High safety integrity

This paper introduces the microprocessor industry support and certification issues. High-level activities to bring safety-critical products to civil and military aviation using Multi-Core Processors (MCPs) are also discussed, based on current Rockwell Collins MCP civil aviation development, with all cores operational, supporting Design Assurance Level (DAL) A. A high-level comparison between automotive and civil/military aviation safety requirements will be discussed. The leading alternative processing architectures are introduced with their history and their vendor’s interest and activities in support of the aviation market. Next steps are described in the areas of MCP certification, alignment of automotive/avionics safety requirements, and potential vendor activities. Finally, our conclusions are summarized.
INTRODUCTION

Avionics vendors are currently at a crossroads where new processing products are offered in SoC and MCP packages and alternative computing architectures are challenging the Power architecture. The initial investment to provide civil-certified avionics products employing MCPs can easily cost more than $10M, which makes processor choice important. The avionics industry is faced with a decision to continue with the Power architecture products, which have reliably served the avionics industry for nearly two decades, or specify potential competitors such as x86 and recent A-core ARM MCPs. The focus of this paper is on medium power (15-45W), medium-high performance processors that provide that performance by using multiple cores.

This paper doesn’t pick a winner as there probably isn’t a single solution for all avionics applications, and at this time there remain many activities underway by MCP vendors that will affect trade studies. This paper does describe the hurdles in bringing a new computing architecture to market in avionics products, and describes alternative MCP vendor’s interest and plans for the avionics market.

MCP SAFETY CERTIFICATION TECHNICAL ISSUES

Faced with little opportunity to increase processing power of Single-Core Processors (SCPs), processor vendors increase performance by providing multiple cores in a single package. When software applications share resources potential resource contention exists that must be solved for deterministic safety-critical systems. This condition is called an interference channel. “Virtual Machine” software partitioning solved application-application resource contention on SCPs in the late 1990s, first used in Rockwell Collins Flight2 and CAAS avionics management systems. Today the use of software partitioning in safety-critical systems is widespread, and this integrity technology is extended to MCPs.

MCP architectures complicate interference mitigation strategies by adding another layer of potential contention (core-to-core) in the architecture. In this case, the MCP vendor provides contention arbitration mechanisms, not the integrator. This means that for safety-critical MCP applications, the integrator must have more intimate knowledge of machine-level arbitration than in the case of SCPs, and that knowledge must be provided by the MCP vendor.

The QorIQ® T2080 internal architecture is shown in Figures 1 and 2 as an MCP example to demonstrate interference channels. The T2080 MCP includes four Power e6500 cores that can be dual-threaded as shown as Thread 1 and Thread 2 (T1 and T2). The source figure is available on the NXP Semiconductor website (Ref. 1).

Figure 1 shows that each of the four e6500 cores has dedicated data and instruction caches. If dual-threading is enabled, an interference channel exists for threads sharing L1 cache memory on the same e6500 core. If dual-threading is not enabled, no L1 cache interference channels exist which simplifies the interference mitigation strategy. However, the L2 cache is shared by the four cores, so an interference channel exists there. Figure 2 shows other areas of potential contention are external memory access through the shared memory controller and Input/Output channels. The Coherency Fabric provides resource access arbitration mechanisms.

MCP software architecture is shown in Figure 3. Three core configurations are shown, and can be implemented on four cores or more in a MCP package.
Application software executes in Virtual Machines (VMs) that are enabled in two ways, 1) virtualized by a hypervisor, which configures and initializes the VMs then recedes into the background, only to respond to exceptions, and 2) by partitioning provided by Guest Operating Systems (GOSs). The hypervisor and GOS products are typically provided by a 3rd-party vendor, and other “foundation software” such as Board Support Packages (BSP) and software drivers are typically developed by the integrator. In this figure the hypervisor and low-layer BSP and driver software are common to all cores, but the integrator can select a Guest Operating System (GOS) that minimizes impact to existing application software, which can greatly reduce application software porting costs.

Other software architecture examples may include an OS that integrates hypervisor and GOS together. While there are advantages to these types of products, they may not support GOS products from other vendors.

Three possible core configurations are shown in the Figure 3 architecture. The Core 0 GOS components support ARINC 653 partitions. The other cores rely on the hypervisor to provide partitioning. The processing space within each partition is statically allocated processing, memory and I/O resources, and is called a Virtual Machine (VM), since the application software running within a partition has no knowledge of any other application executing on the same core. The Core 1 example shows a single partition with a General Purpose GOS, such as Linux, as an example. Core 2 configuration has a separate GOS in each of the partitioned VMs.

Rockwell Collins first provided software partitioning in single-core processors nearly 20 years ago on Flight2/CAAS Avionics Management Systems. While partitioning improves software modularity and coupling (good things), the real motivation for partitioning is reducing certification costs of application software through hard separation and guaranteed availability of required resources (memory, space, time, and I/O resources).

SAFETY-CRITICAL MULTI-CORE PROCESSOR DEVELOPMENT DESCRIPTION

The cost and schedule required to develop a MCP for aviation depends on the intended usage. For civil avionics a Line Replaceable Unit (LRU) is certified by achieving a Technical Standard Order (TSO), or a subsystem can be part of an overall aircraft Type Certification (TC) or Supplemental Type Certification (STC). Military Air Worthiness Release (AWR) is granted by the service’s airworthiness authorities at the LRU, system or aircraft level.

Civil and military originating safety requirements are typically applied at the functional level using the DO-178C Design Assurance Levels (DAL) designations A-E, which consider the effect of a failure (hazard level) or the potential for presenting false or misleading information to the aircrew, or otherwise operating in an unsafe condition. DAL A is the highest designation, with potential for catastrophic effects, while a DAL E assignment means that little or no potential exists of affecting aircraft or passenger safety in the event of abnormal system operation.

Systems that are assigned at high DAL, such as A or B, require more developmental rigor and therefore will cost...
more and take longer to bring to market than systems assigned lower safety criticality.

As shown in Figures 1 and 2, SoCs and MCPs are much more complex than partitioned single-core devices due to increased contention for shared resources such as cache memory and system memory, I/O and internal communications. Complex SoCs require additional safety engineering rigor, and MCPs have other additional considerations for safety-critical applications.

Traditional design process documents in use for civil and military aviation, such as DO-178C and DO-254, were written before MCPs were available and therefore do not address the certification of systems using complex MCPs.

To address these considerations the FAA has prepared and published guidance for use of MCPs in safety-critical aviation applications, called the Certification Authorities Software Team (CAST) Position Paper CAST-32A (Ref. 2). Military aviation airworthiness authorities have already incorporated DO-178C and DO-254 process requirements and will most likely follow the FAA lead for MCPs used in military aircraft.

CAST 32A states that the document does not apply to certain MCP system designs that allow dynamic operation features such as re-allocating software applications to different cores during operation. In general, determinism of systems using dynamic features is difficult to bound and prove, and safety-critical processors generally configure resource allocations statically at system start. Safety engineering guidance is provided in the form of the following objectives (Ref. 2) along with our commentary. FAA certification requires evidence that these objectives are met.

**MCP_Planning_1:** Plans identify MCP software architecture (including dynamic and IMA* aspects)

*Integrated Modular Avionics

**Importance:** Provides the overall system design context for the certifying authority

**MCP_Planning_2:** Plans provide a high-level description of how MCP shared resources and dynamic features will be used and how the applicant intends to allocate and verify the use of shared resources

*Importance: Shows the applicant has thought through critical multicore issues that could impact the execution of the application software

**MCP_Resource_Usage_1:** The applicant has determined and documented the MCP configuration settings

*Importance: MCP’s are extremely complex and likely have configuration settings that could negatively impact system safety

**MCP_Resource_Usage_2:** The applicant has planned, developed, documented, and verified a means that ensures that in the event of any of the Critical Configuration Settings of the MCP being inadvertently altered, an appropriate means of mitigation is specified

*Importance: Configuration settings could be inadvertently modified by software errors or single event upsets in ways that result in undefined behavior if not mitigated

**MCP_Resource_Usage_3:** The applicant has identified the interference channels and has verified the means of mitigation of the interference

*Importance: Interference channels are a source of jitter and performance degradation and may have significant, negative impact on the determinism of the processing system

**MCP_Resource_Usage_4:** Identification/allocation/verification on the target processor of the available resources of the MCP and of its interconnect

*Importance: MCP embedded interconnect mechanisms and shared resource arbitration are too complex and are not documented well enough to accurately model for host-based verification

**MCP_Software_1:** Verification that all the hosted software components function correctly and have sufficient time to complete their execution when all the hosted software is executing in the intended final configuration

*Importance: Software applications running simultaneously on different cores impact each other’s execution timing and need to be integrated together to understand the impacts to operational behavior

**MCP_Software_2:** Verification that the data and control coupling is correct during software requirement-based testing

*Importance: Data and control coupling across cores is more complex than coupling across partitions on a single-core processor and may result in unintended behavior if not verified to be correct

**MCP_Error_Handling_1:** Identification of the effects of failures that may occur within the MCP and plan, design, implement and verify means by which to detect and handle those failures in a fail-safe manner

*Importance: The high level of integration of device functions and peripheral interfaces typically found within multicore system-on-chip designs drives a need for more built-in-test and monitoring for desired behavior

**MCP_Accomplishment_Summary_1:** The applicant has summarized in their SAS, HAS or other deliverable documentation how they have met each of the objectives of this document
Importance: Provides a reference for the evidence developed to build assurance that the MCP system design is appropriate for safety critical use

Figure 4 identifies MCP certification high-level tasks and workflow. The process of preparing an MCP for safety-critical aviation application is a combination of traditional certification tasks and MCP-specific activities included to satisfy the CAST 32A objectives. This process was successfully used to develop our Power-based T2080 MCP for use in civil aviation applications. These workflow activities will be described in subsequent paragraphs.

This general methodology will need to be applied for the alternative architectures, so this process demonstrates what work is required to bring these alternative architectures to the safety-critical aviation market.

In Figure 4 steps 8 and 12 are shown to be iterative. Known and potential interference channels are identified in Step 8, but other interference channels may be discovered in steps 9-12 that could cause reiteration back to Step 8.

1. Define and document intended MCP use
System design identifies potential safety hazards and hazard severity to aircraft and passenger safety in the event of unexpected system operation. These hazards can be eliminated or mitigated in the system design process, and functions are assigned criticality based on this system design. The hardware and software that provide this functionality are also assigned criticality, generally in the form of DALs. For safety-critical hardware and software, the cost and development schedule increase as the assigned DAL approaches the highest criticalities (DAL A and B) because DO-178C, DO-254, and now CAST 32A identify addition development activities for DAL A and B-assigned systems. Therefore, it is important to document the intended MCP use as usage dictates the subsequent activities, cost and schedule of MCP development.

2. Select MCP Vendor
MCP vendor selection for safety-critical applications extends beyond processing performance and power consumption, including the following four factors:

Vendor commitment and long-term support to the aviation market
Aviation product vendors will invest 10s of millions of dollars on MCP safety engineering to bring an MCP architecture into the aviation market. In addition, attaining product certification with civil or military airworthiness authorities can take 3 or more years. To be selected, a MCP vendor must demonstrate strong support, specifically for the aviation market. Aviation products tend to be long-lived, with product life-cycles stretching to 20 or more years in some cases, requiring longer product availability than consumer and server-based products.

Figure 4: Processor safety design process workflow

Mature and rigorous manufacturing processes
In general, rigorous manufacturing processes are key to achieving required yields, so this factor is generally met by MCP vendors. In addition, timely publication of errata is required.

Technical data sharing
For high-integrity applications, an avionics vendor will require potentially proprietary information to complete MCP safety engineering. The MCP architecture must be understood to determine potential non-deterministic MCP
behavior (e.g., interference channel). Vendor support may be required to develop mitigation strategies to achieve the required level of determinism.

**Processing platform maturity**
MCP vendor-provided data is analyzed to determine if the MCP is mature, and to determine how the MCP vendor is supporting integrators by correcting reported problems or providing work-arounds to known issues that is described in published errata.

3. **Audit vendor development and production processes**
Civil MCP certification will benefit from a vendor audit of production and configuration management processes.

4. **Perform hypervisor and operating system vendor(s) source selection study, develop strategic relationships**
As with MCP vendors, the relationship between integrator and Hypervisor and OS vendors is critical. During safety-critical MCP development, safety requirements will be allocated, by the integrator, to the hypervisor/OS vendors. Generally even Off-The-Shelf (OTS) software products require some modification, particularly if the MCP is a new design. For civil and military use, these products will be developed in accordance with the specified DAL.

5. **Analyze MCP vendor-supplied technical data**
The MCP vendor will provide technical data in the form of training, processor documentation, embedded micro-code, register usage, and errata. The purpose of this analysis is to gain intimate understanding of MCP operations, and to determine if sufficient technical data is available to perform Steps 8-10. MCP register usage is needed to develop the best MCP configuration for the desired usage.

6 & 7: **Develop Prototype HW and SW to support testing and characterization**
Prototype hardware and software is required to perform MCP performance testing and characterization. In some cases vendor-provided development boards may be used. In aviation, application-unique peripheral devices may be reused to leverage existing certification artifacts, requiring a custom prototype design. The design should be close enough to the final configuration to make the collected metrics meaningful.

8. **Identify known and potential interference channels within the MCP**
Interference channels are identified referencing Figures 1 and 2. Additional vendor-provided technical data may reveal additional interference channels once contention-arbitration mechanisms and other MCP design details are discovered. It is the responsibility of the avionics component supplier to prove that the known interference channels, as well as all other MCP designs, operate as intended by the MCP vendor.

9. **Define “worst-case performance and determinism bounding for our domain”**
For safety-critical MCP applications software functions must complete in a deterministic manner. Operationally, that means that the application cannot depend on MCP performance that is not always available. To ensure that performance, MCP analysis must be conducted that determines the worst-case performance that is possible, and bound the performance available to applications with worst-case metrics. In this step we are essentially trading potential performance with always-available deterministic performance.

10. **Interference channel testing and characterization**
In this step, exploratory interference channel performance testing is conducted, using the prototype hardware and test scenarios created for all known interference channels, in order to develop an adequate understanding of the interference profiles. This sub-process is typically iterative, where testing yields performance metrics, analysis is conducted, and system configuration adjustments are made in order to achieve the performance requirements identified in Step 9. An example of this process is tuning the cache configuration to balance performance and deterministic system behavior. Oftentimes additional, previously unidentified interference channels are discovered by analysis test results. The sub-process continues, iteratively modifying test scenarios as needed, until all test scenarios are executed and a final system configuration is achieved.

11. **Develop hardware and software mitigation strategies to meet CAST 32A guidance and objectives**
To meet CAST 32A objectives a risk mitigation strategy is developed to provide an acceptable level of determinism. Strategies can include safety nets, statically-set processor configurations, and performance bounding. Civil certification authorities require documented evidence that the MCP objectives are met.

12. **Develop low-level, safety-related hardware and software requirements**
Safety requirements are formally instantiated for all configuration settings and mitigation techniques identified during interference channel analysis.

13. **Develop final system configuration**
This stage advances the MCP hardware and software from a prototype to the final configuration by designing to low-level safety requirements developed in stage 12.

14. **Test final system configuration to safety requirements**
Hardware and software test procedures are developed to the requirements developed in all previous stages.

15. **Process Audits**
FAA civil certification processes (DO-178C, DO-254) require four thorough Stages Of Involvement (SOI) audits of evidence provided by the developer to satisfy safety authority objectives.

Our MCP design that is in the final safety certification phases uses the Power QorIQ T2080 and is certifiable for other applications, with available certification artifacts, determinism analysis, and integrator tools that prove the MCP can be operated in a safe and deterministic manner.
Processor circuit cards with integrated third-party OS and Hypervisor products that have been developed to DO-178C process requirements may be called “certifiable”. If the CAST 32A objectives aren’t met and the 15 process steps are not complete for the “certifiable” product, full safety certification will require additional development, potentially causing a complete redesign of hardware and software.

**AUTOMOTIVE SAFETY AND INDUSTRIAL AUTOMATION POTENTIAL BENEFIT TO AVIATION**

US DOT current policy, shared with the National Highway Traffic Safety Administration (NHTSA) is provided as non-regulatory guidance that is intended to not interfere with automotive technical advancements (Ref. 3). Automotive manufacturer’s compliance with this guidance is voluntary, which is in contrast to policies of the FAA, which is interestingly another US DOT agency.

FAA integrity guidance is described in DO-178C and DO-254 safety process requirements. ISO 26262 provides guidance for automotive use and IEC 61508 provides guidance for general industrial applications. Initial comparisons between the three standards indicate sizable gaps to bring ISO 26262 products into alignment with DO-178C and DO-254 requirements. ISO 26262 compliance alone will not be acceptable for products intended for aircraft falling under FAA civil certification rules. That said, there is interest in reusing safety evidence created for industrial or automotive applications in avionics applications.

Table 1 is a summary comparison of criticalities defined in three safety standards of interest to aviation (Ref. 4). The standards are ISO/IEC 61508, ISO 26262, and DO-178C/DO-254. The first two standards address failure rate limits, while aviation standard also addresses the probability of a system providing false or misleading information to the aircrew. As shown, IEC 61508 and ISO 26262 criticality assignments (SIL/ASIL) do not map to DO-178C DAL A. This indicates that DO-178C DAL A requires a higher level of integrity than IEC 61508 or ISO 26262 currently address. Additional analysis was undertaken to determine what safety evidence could be reused from IEC 61508 and ISO 26262.

<table>
<thead>
<tr>
<th>Fail Rate (prob/hr)</th>
<th>ISO/IEC</th>
<th>ISO 26262</th>
<th>DO-178C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^-5 - 10^-6</td>
<td>SIL1</td>
<td>ASIL A</td>
<td>DAL C</td>
</tr>
<tr>
<td>10^-6 - 10^-7</td>
<td>SIL2</td>
<td>ASIL B</td>
<td></td>
</tr>
<tr>
<td>10^-7 - 10^-8</td>
<td>SIL3</td>
<td>ASIL C</td>
<td>DAL B</td>
</tr>
<tr>
<td>10^-8 - 10^-9</td>
<td>SIL4</td>
<td>ASIL D4</td>
<td></td>
</tr>
<tr>
<td>&lt; 10^-9</td>
<td></td>
<td></td>
<td>DAL A</td>
</tr>
</tbody>
</table>

To that end, we commissioned a TSO Compliance Representatives (TCR) to perform an initial mapping between the DO-178C Software Lifecycle Process Activities and the applicable ISO 26262 activities defined in Part 6 (Product development at the software level). A more detailed analysis is ongoing, but for this paper our initial results are provided showing relative gaps in comparisons.

Several activities in the Software Planning and Software Development processes were identified that are covered or partially covered by analogous activities identified in ISO 26262, part 6 (and other parts where applicable); and we also identified several activities that are not covered. Preliminary results for DAL A coverage are shown in Figure 5.

![Figure 5: Preliminary ISO 26262 Fulfillment of DO-178C DAL A Software Life Cycle Activities](image)

We also made comparisons for DAL B and C. For brevity, ISO 26262 - DO-178C DAL B and C coverage charts are not shown. But as expected, as the DO-178C DAL is reduced from DAL A to DAL B and C, the activity coverage provided by ISO 26262 guidance increases. Partial coverage credit was given for some ISO 26262 activities, which explains the fractional coverage shown in Figure 5. This graph is to be used for estimation of effort only.
CANDIDATE ALTERNATIVE ARCHITECTURES

Two candidate architectures support the large automotive market and vendors of these two architectures are currently exploring or developing the business case to enter the aviation market. We also considered the Risc-V architecture for this study. While interest in this architecture is increasing from commercial industries, academia, and government services, we discounted it due to a current lack of silicon options (CPU).

X86
These architectures are offered by two vendors, Intel® and Advanced Micro Devices®. Intel had a strong presence in military aviation in the 1980s and 1990s but left the market to the Power architecture which emerged from the Apple-IBM-Motorola alliance.

Intel currently supports the industrial automation and automotive industries with processor products from its Internet of Things Group (IoTG). IoTG has identified specific MCPs from Intel’s product roadmap as Functional Safety (FuSa) capable and are being handled through the Intel FuSa lifecycle. The FuSa lifecycle represents the milestones, activities, and work products that enable development of FuSa capable products. IoTG is committing to providing extended temp and 10-15 year availability for these FuSa-specific devices.

Intel is initially offering FuSa architectures in a high-performance Xeon®-DE package. It include on-chip cores, cache memory, cryptography and significant I/O processing. As a leader in the server processor market, the strength of the X86 architecture is performance and Intel claims industry-leading performance per watt.


Advanced RISC Machine (ARM)
ARM is a family of processors designed by ARM Holdings of Cambridge, UK, which licenses the designs to manufacturers and provides development tools. As such, the internal workings of the ARM processors can be well-understood with a financial investment. This access is critical to understanding potential and known resource contention issues for multi-core processor use in aviation applications.

ARM cores are available in a variety of packages, from a multi-core processor (MCP) to a System On Chip (SOC), which may include processors, memory, 3D graphics, hard coded peripherals, and programmable logic. In addition, some ARM vendors have taken some design license with the base IP, so there may be limited ability to reapply certification processes and artifacts between vendors or packages.

TECHNICAL COMPARISONS TO POWER ARCHITECTURE

Each architecture (Power, X86, ARM) is available in a number of configurations, enabling vendors to meet a range of computing and power requirements within their product lines. In addition, some products offer built-in accelerators that may or may not be of value for a given application. Therefore this comparison does not select a specific product from each product line for comparison. This discussion is broader in nature and the results are general tendencies of the product lines, rather than a comparison of specific products.

The Power and ARM Architectures are based on a Reduced Instruction Set Computer (RISC) design while the X86 processors are Complex Instruction Set Computers (CISC). This is really a simplification, since Power and ARM A-core processors use some complex operations, so they can’t be considered a pure RISC design. However for the purposes of this discussion Power and ARM have a simpler instruction set than x86.

All three architectures (Power, X86, ARM) contain protected Intellectual Property. For high-integrity use, knowledge of internal processor information is required to understand where resource contention exists that can affect deterministic operation. For Power and X86, this IP is considered to be a trade secret and access to IP is dependent on the vendor. ARM IP is also protected but can be purchased. Recent high-end ARM cores are approaching the performance of Power cores (e6500), particularly if the e6500 accelerators are disabled for high-integrity use.

CISC processors, such as X86, perform multiple operations with a single instruction. Compared to a RISC processor, additional hardware is included in each core to accomplish the complex instruction. As a result the compiled instructions are more compact, requiring less code memory for a given function. In general, the additional transistors required to support CISC will also require additional power. That said, Intel has invested heavily in reduced transistor feature size which mitigates some of the power consumption differences.

Pure RISC processors (Power, ARM) have a minimal set of instructions that operate in a single clock cycle. This simplicity reduces the number of transistors required to support the instruction set but more instructions are required to perform a given function, compared to a CISC design. As a result, code space memory size will be higher for a pure RISC application, compared to a similar CISC application. The RISC architectures lend themselves to safety-critical applications, because processor characterization of deterministic behavior is easier to accomplish with a simpler design. With fewer transistors, RISC processors tend to be more power-efficient than CISC processors. ARM-based products have been extensively used in battery or otherwise low-powered devices such as smartphones.
Of the three non-deterministic architectures (Power, ARM A-cores, X86), only Power products have been widely accepted for use in safety-critical avionics applications. Therefore the alternative architectures may face additional scrutiny by aviation safety authorities. Deterministic ARM R-core and M-cores have seen use in less computation-intensive safety-critical applications, but are not suitable for General Purpose Processing and are therefore not considered in this paper.

The differences between the three architectures are summarized in Table 2. Again, these are not a comparison of a specific Power product against a specific X86 product against a specific ARM product, but are a relative comparison of the overall product lines and their characteristics.

Table 2: Relative comparison of processing architectures

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Power</th>
<th>x86</th>
<th>ARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction</td>
<td>RISC*</td>
<td>CISC</td>
<td>RISC</td>
</tr>
<tr>
<td>Computing Power</td>
<td>Medium</td>
<td>High</td>
<td>Low-Medium</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>Low-Medium</td>
<td>Medium-High</td>
<td>Low-Medium</td>
</tr>
<tr>
<td>IP Access</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Processor Complexity</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
</tbody>
</table>

* With performance extensions

For avionics applications where card edge-conduction cooling is employed, 45 watts is a typical upper power limit for devices and is considered “high power” in this context. Medium power is considered 15-25 watts, and low power is less than 15 watts.

**CURRENT STATUS OF THE WORK**

We are currently meeting with X86 and ARM vendors that have expressed interest in the aviation market. These vendors are evaluating the aviation market and determining what business cases can be made for entry. These are not “whether” decisions, but rather “how” decisions.

There is interest in understanding what artifacts developed for the automotive industry can be leveraged into the aviation industry. Current work in progress is comparison of ISO 26262 (including the recent ISO/FDIS 26262-11:2018, with specific guidelines for application of FuSa, enabling state-of-the-art semiconductor technologies and RTCA DO-254 documents. These evaluations are under development both by potential MCP vendors and aviation integrators, including Rockwell Collins.

Regarding the progress of MCP use in safety-critical avionics for TSO, we are completing our final FAA safety process audits. Our customer is currently performing test flights with the product. This will be the first time a TSO will be granted to avionics using a MCP in a DAL A application, with all cores driven simultaneously. The MCP processes developed for this Power architecture will be applicable to either X86 or ARM in the future.

**SUMMARY**

PC and server-class processor computing performance is achieved by integrating multiple processing cores into a single package. The multiple cores share some level of resources that affect determinism while generally providing higher performance. Military and civil safety certification has become more complicated as a result of MCP development because of reduced determinism and potential contention.

The US FAA has published a position paper for product certification using MCPs, and safety certification requires showing evidence of meeting the objectives included in the position paper.

Avionics MCP integrators provide high-integrity products to military and civil aviation. Compared to SCPs, MCP integrators are more reliant on vendor-provided MCP information to understand internal resource contention (interference channels) and how the MCP arbitrates contention. Often this intimate information includes trade secrets and is not released to the public, so access to this information can add risk to a MCP product development.

At least two MCP vendors are considering entrance to the high-integrity avionics market, offering X86 and ARM architectures. While Power architecture products are still viable and supportable for new designs, ARM and X86 will be considered to be viable long-term alternatives.

ARM and x86 are currently providing products to the automotive market from two diverse directions, X86 from desktop/server markets where processors are selected on computing performance, and ARM from the embedded/battery-powered market, where low power consumption is paramount.

ARM and x86 currently support the automotive industry which has increasing need of high-integrity operation, longer product availability periods, and extended temperatures, all desirable for the aviation industry. In addition, the automotive industry has developed safety guidance including specific guidelines for application of FuSa, enabling state-of-the-art semiconductors technologies in that industry. MCPs that are compliant with that guidance may have safety-design artifacts that are of value to military and civil aviation safety certification. Initial comparisons between automotive and aviation safety requirements indicate some reuse is possible but further investigation is required to fully understand the reuse potential.

The first civil-certified avionics products using Power MCPs, using all cores, supporting DAL A, are now becoming available. These are the test cases for MCP DAL A certification and meeting the FAA MCP integration objectives. Safety certification of alternative processing architecture products will require the same process.
The experience gained in Power MCP certification is directly applicable to safety certification of alternative MCPs.

REFERENCES

1. “QorIQ T2080 Communication Processor, NXP Power Architecture® Processors”, published by NXP Electronics, accessed 2/16/2018

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


   https://pdfs.semanticscholar.org/02b1/19e488c4dd1132f352f41b8ebe8b6b7a9fb5.pdf