Conference Paper

Towards the Certification of Multicore Platforms in the Avionics Domain

Muhammad Ali Awan

Patrick Meumeu Yomsi

Konstantinos Bletsas

Vincent Nélis

Eduardo Tovar

Pedro Souto

CISTER-TR-150716

2015/03/24

Towards the Certification of Multicore Platforms in the Avionics Domain

Muhammad Ali Awan, Patrick Meumeu Yomsi, Konstantinos Bletsas, Vincent Nélis, Eduardo Tovar, Pedro Souto

CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: muaan@isep.ipp.pt, pamyo@isep.ipp.pt, ksbs@isep.ipp.pt, nelis@isep.ipp.pt, emt@isep.ipp.pt, pfs@fe.up.pt

http://www.cister.isep.ipp.pt

Abstract

In the last decade, the semiconductor industry has experienced a paradigm shift from single core design to a multicore architecture era. This trend is driven by the fact that the increase in clock speed to enhance the performance of a core has hit a limit as the performance per watt became costly at high frequencies. A multicore processor (MCP) combines two or more cores into a single package (single or multiple dies) such that they can execute programs simultaneously. Although this feature is very appealing, it comes with new challenges, unfortunately. Most MCP platforms present many sources of unpredictability as the large majority of hardware vendors are mainly interested by improving the average performance of the system. This work discusses some sources of non-determinism and highlights the envisioned methodology to mitigate or eliminate their effect in the context of safety critical systems.

Towards the Certification of Multicore Platforms in the Avionics Domain

M. Ali Awan[1], P. Meumeu Yomsi, K. Bletsas, V. Nélis, E. Tovar, and P. Souto[2]

[1] CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto
{muaan,pamyo,ksbs,nelis,emt}@isep.ipp.pt
[2] University of Porto, FEUP-Faculty of Engineering
pfs@fe.up.pt

Motivation. In the last decade, the semiconductor industry has experienced a paradigm shift from single core design to a multicore architecture era. This trend is driven by the fact that the increase in clock speed to enhance the performance of a core has hit a limit as the performance per watt became costly at high frequencies. A multicore processor (MCP) combines two or more cores into a single package (single or multiple dies) such that they can execute programs simultaneously. Although this feature is very appealing, it comes with new challenges, unfortunately. Most MCP platforms present many sources of unpredictability as the large majority of hardware vendors are mainly interested by improving the average performance of the system.

In the avionics domain, the need for extra functionality demands to deploy safety-critical applications on more than one core in order to cope with the SWaP2-C2 (Size, Weight and Power, Performance, Cooling and Cost) constraints. Nonetheless, the certification authorities are very skeptical about adopting MCP platforms. This position is justified by the inherent non-deterministic behaviour of MCP platforms which can impact the safety, the performance and the integrity of the airborne system. The airspace industry is currently addressing the certifiability of MCP platforms. In the recently published "CAST-32" position paper [1], the certification authorities from North and South America, Europe and Asia have expressed their concerns about the use of a *two-core* platform for safety-critical avionics systems. They identified sources of non-determinism in such a platform. On that front, CISTER researchers have taken one step forward to handle the issues raised in that paper through projects such as EMC^2. Some sources of non-determinism are discussed below and the envisioned methodology to mitigate or eliminate their effect is highlighted. A focus on the development of new analysis techniques to determine upper-bounds on the interference resulting from resources sharing is favoured, at both the hardware and software levels.

Sources of non-determinism and envisioned methodology.

Caches. MCP platforms share cache(s) among tasks running on the same or different cores. The effect of shared caches on the worst-case execution of the tasks has been pointed out. Assuming a preemptive scheduler, tasks running on the same core may evict the cache lines of another task (preempted) even if the cache has been partitioned at core-level. Indeed, the preempted task has to suffer a penalty by reloading its data prior to resuming execution. This penalty increases the execution time of the task. In order to guarantee the schedulability of safety-critical application, real-time schedulers require an upper bound on the worst-case execution time (WCET) of each task. Hence, WCET tools must consider the effect of preemption on the same core in addition to the interference from other cores on the shared cache(s). This is a non-trivial exercise as the preemption instances are non-deterministic and each core is not aware of what is being executed on the other cores.

Envisioned methodology. One possible solution to circumvent the above issue is to partition all the shared caches among the cores and handle the cache-related preemption delays (CRPD) locally on each core. To this end, we intend to exploit the *cache usage profiles of each task* and integrate the effect of the CRPD to the WCET tools.

Memory subsystem. Tasks running on different cores may interfere while accessing the shared memory. A task (say x) running in one core (say core A) may need to access the shared memory to fetch data and gets its request blocked by that of another task (say y) executing on another core (say core B). This situation may lead to a stall of core A, thus increasing the WCET of x. The effect of this contention should be included in WCET tools. Core A may also be "locked" accessing some shared memory locations, thus causing extra delay in the WCET of x. This may lead to unexpected system behaviours.

Envisioned methodology. We plan to elaborate solutions to handle the shared memory such that the execution of the tasks on the target MCP platform is free of race conditions, data starvation, deadlocks or live-locks.

I/O subsystem and interconnect features. Most avionics and control system applications require I/O devices to perform some functions. These devices are also shared among different cores on MCP platforms. The interaction between the cores and such a device is performed through different mechanisms such as "programmed I/O", "direct memory access (DMA)" and "interrupts". In addition, MCP platforms use different types of interconnects for resource sharing. This in turn makes the timing analysis difficult to assess as the I/O accesses may affect both the traffic between the cores and the memory, and the caches coherency.

Envisioned methodology. In order to ensure the temporal correctness of the system, we intend to analyze the interference of the DMA traffic on various relevant components.

Acknowledgements. This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and when applicable, co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within project UID/CEC/04234/2013 (CISTER Research Centre); also by FCT and the EU ARTEMIS JU within project(s) ARTEMIS/0001/2013 - JU grant nr.621429 (EMC2).

References

1. CAST-32: Multi-core processors. In: Position Paper (2014)