Formal Verification of Temporal Predictability in Real-time Systems

- **Deadline:** April 30, 2019
- **Career levels:** PhD student
- **Keywords:**

A PhD position is available at CEA Paris-Saclay within the DACLE division (Architecture, IC Design & Embedded Software) in a team working on real-time systems. A joint scientific supervision will be realized with Telecom ParisTech. More information about CEA LIST can be found here.

Context: Modern computer architectures are designed to alleviate the bottleneck between processors, and memory systems, leading to utilization of caches, (automatic) pre-fetching, branch predictors, pipeline. Such complex architectures are also used in real-time system design to cope with the continuous increase in complexity and in performance requirements of software, in particular of so-called mixed-criticality systems, mixing tasks of different levels of criticalities. Real-time tasks of a mixed-criticality system require to satisfy, a posteriori, stringent timing behavior under all conditions, i.e. in the worst-case scenario. To this end, a worst-case analysis is first performed on each task, so that Worst-Case Execution Times (WCETs) are estimated. A schedulability analysis then combines together these WCETs, and the contention overheads associated due to the parallel execution with all the other tasks, in order to check whether timing constraints can be fulfilled or not.

The WCET analysis computes sound and (desirably) tight worst-case execution bounds, exploring, via convenient abstractions, all the execution paths of a program running on a computer architecture. This searching process is however complicated by the presence of timing anomalies [1]. In essence, a timing anomaly is a counterintuitive behavior in the sense that the local worst-case timing behavior does not result in the global worst-case performance [2]. In a real-time system exhibiting timing anomalies, a worst-case analysis must analyze all potential executions of the real-time program, considering all potentially feasible hardware states. As hardware components improve the average execution time of programs and not their WCETs, the number of hardware states to explore in these analyses increases in modern computer architectures.

Problem and Related Work: This large number of states to be explored by WCET analysis tools leads to a natural compromise between analysis precision and computational overhead (or even feasibility). WCET analysis tools consequently operate on an abstract model of the actual behavior of both, software and hardware. However, these models are currently informal models, preventing the exploration of both hardware and software patterns to evaluate the predictability issues of existing systems or to guide the construction of predictable mixed-criticality systems by feeding these patterns to AI-based algorithms.

The automatic detection of timing anomalies is addressed in [3] by building a prediction graph which is a compact representation of instruction-level simulations of program paths. [4] uses bounded model checking to explore the abstract architecture state space. A sequence of instructions that generates a scheduling timing-anomalous behavior is built, enabling the identification of timing anomalies independently from a given code. However, no complexity analysis or runtime performance results are reported. Besides, the specifics of the formal models are not described. Lastly, [5] addresses the problem of timing anomalies with sound techniques, ranging from pipeline stalls at specific points of the hardware state [6] to over-approximation of local effects with integer linear programming. However, this work does not target mixed-criticality systems and no formal models are used.
Goals and Expected Contributions

In this PhD proposal, we target the design of formal models of hardware components of computer architecture and of their Instruction Set Architecture (ISA) to identify code-specific timing predictability issues [7]. The following two contributions are expected of this PhD thesis:

- **the design of appropriate abstractions** for a feasible identification of code-specific timing predictability issues, a major concern as most commercial architectures are non-predictable. Automatic detection of anomalous timing behaviors is useful to later insert mitigation mechanisms in order to support the design of predictable systems.
- **the exploration of the trade-off** between average and worst-case performances in order to support the efficient execution of mixed-criticality real-time systems over multicore architectures. Both software and hardware mitigations will be considered, such as the definition of appropriate compilation and/or scheduling rules or adapting the behavior of the hardware to the type of task currently being executed, as in [8].

As use cases, several hardware architectures will be considered, such as the predictable platform Patmos [9], but also close to COTS architectures such as RISC-V based processors.

**Candidate Profile and Application**

- M.Sc. in Computer Science
- Knowledge in: computer architecture, real-time systems, compilers, formal methods
- Programming, preferably including HDL (Verilog, VHDL), functional languages, scripting
- Excellent written and spoken English
- Communication and writing skills
- Teamwork motivation and autonomy

Please submit the following information: detailed CV, motivation letter, contact information of the persons that can provide recommendations for you and any scientific documents written by you (publications, master thesis etc). Feel free to ask questions / request more information before applying!

**References**