MINOTAuR: A Timing-Predictable Open Source RISC-V Core Featuring Speculative Execution

Alban Gruin* Thomas Carle Christine Rochange Pascal Sainrat

name.surname@irit.fr
IRIT — Université Toulouse 3 — CNRS

1 Motivation

Worst-Case Execution Time (WCET) analysis is required to be able to schedule real-time tasks in such a way that they meet their deadlines. In multicore processors, interference between cores (e.g. on a shared memory bus) impacts the WCET, and must be taken into account.

The usual solution is the compositional approach: latencies to shared resources, including the effects of interference, are analysed separately, then combined to the WCET of tasks. However, so called "timing anomalies" make this approach unsafe. Timing anomalies are situations in which a worst local situation (e.g. a cache miss) does not result in the worst global situation (i.e. the WCET). To allow compositionality, one needs to guarantee that the processor is free from timing anomalies. One solution is to design a "monotonous" pipeline, in which an instruction cannot delay the progress of an older instruction in the pipeline.

Several processors have been designed with timing-predictability in mind. One of the most recent efforts is the Strictly In-Order (SIC) by Hahn et al. [2], a simple 5-stage in-order processor. They proved the monotonicity of the core with a formal logic framework. In practice, the fetch stage is prevented from accessing the memory bus while a memory instruction is pending in the pipeline, and the core is unable to speculate instructions.

2 Contributions

In [1], we introduce MINOTAuR (Mostly IN-Order Timing predictAble processoR), a predictable RISC-V processor that is a monotonous version of the open-source Ariane core [3]. This processor features more advanced execution mechanisms, such as branch prediction, instruction queues, and a scoreboard. In order to reconcile performance and timing-predictability, we modified the core to make its pipeline monotonic, while retaining the speculative execution capability. We proved the correctness of our solution by re-using and extending the formal framework pioneered by Hahn et al.

Our work shows that established designs can be modified to be made timing-predictable. In this, the open nature of RISC-V and its ecosystem allows us to appropriate existing components (toolchain, cores, etc.), and make our changes open, so that the community can immediately benefit from them.

References

[1] Alban Gruin, Thomas Carle, Hugues Cassé, and Christine Rochange. Speculative execution and timing predictability in an open source RISC-V core. In *2021 IEEE Real-Time Systems Symposium (RTSS)*, pages 393–404. IEEE, 2021.

[2] Sebastian Hahn and Jan Reineke. Design and analysis of SIC: A provably timing-predictable pipelined processor core. *Real-Time Systems*, 56(2):207–245, 2020.

[3] ThalesGroup. CVA6-softcore-contest. `https://github.com/ThalesGroup/cva6-softcore-contest/tree/0abb1a6`.

*Presenter