Worst-case Execution Time Analysis for Fault-Tolerance Design on RISC-V

Nowadays dependability becomes a critical challenge for all computing systems, especially embedded systems. Due to universe radiation and particle strikes, transient faults may happen and flip bits in underlying hardware. Since the rate of fault is often expected to be low (currently), instead of over-provision with harden components statically, software fault tolerance techniques are widely-used due to their flexibility and portability. However, their success is often at the cost of time overhead, which is critical to the real-time systems.

The assumption of how fault-tolerance designs is often too pessimistic from the modeling perspective. In fact, the practice of countermeasures against faults can be down to a finer granularity, e.g., at the instruction level. For example, advanced architectures like RISC-V have imposed a great potential for this level. However, classical worst-case response time analysis techniques haven’t taken this fact into account, and certainly some efforts must be paid here. This assignment is expected to enhance an existing WCET analyser to mitigate the unnecessary pessimism when fault-tolerance techniques are applied practically. Depending on the status of the chosen WCET analyzer, RISC-V relevant state-of-the-arts can be further considered.

The assignment includes the following steps:

1. Understanding WCET analyzers, i.e., OTAWA, and state-of-the-art fault-tolerance techniques
2. Analyzing program structures and identifying effective portions / control flows to be protected
3. Enhancing the existing WCET timing analyses by modeling the fact of fault-tolerance techniques
4. Evaluating the tightness of the derived WCET bound by comparing with relevant studies

*Other suggestions and related topics are also welcome.*

Contact: Kuan-Hsun Chen and Marco Ottavi