Analyzing fault parameters triggering timing anomalies

Topic	Fault trees, formal methods, critical systems
City and country	Grenoble, France
Team in the lab	Shared Resources
Advisors	Lionel Rieg, Catherine Vigouroux, Mathias Ramparison — first.last@univ-grenoble-alpes.fr

General presentation of the topic In planes and cars, any failure may lead to injuries for driver or passengers. In order to understand and mitigate these failures, it is important to analyze what their causes are, and which combination of elementary events may trigger them. One such tool is *attack-fault trees*, a model used to express logical relations between atomic events, possibly time-dependent, that might lead to an undesirable event. Attack-fault trees are used for instance to understand causes and consequences of apparently independent events and mitigate consequences by helping e.g., to find counter-measures to vulnerabilities in cyber-security, or to improve safety of cyber-physical systems in the aeronautic and space industries. Through translation to *timed automata* with unknown timing constraints, called *parameters*, attack-fault trees benefit of the powerful framework brought by the parametric timed automata theory and can be used to synthesize timing constraints under which a failure/breach occurs.

We want to use them to model and understand a surprising phenomenon of processors, called a *timing anomaly*. A timing anomaly happens whenever a local worst-case behaviour (a cache miss, for instance) leads to a shorter global execution than the local best case (a cache hit may cost more). In the case of a cache hit, this may happen if the access is terminated earlier and a time window is free for executing something else in the pipeline that will impact the total execution time.

Objective of the internship The objective of this internship is to explore the type of processors that creates timing anomalies using the fault tree methodology. First, we want to identify the precise combination of conditions under which timing anomalies may occur using fault trees. Then, converting them to parametric automata, we want to synthesize which parameter values lead to timing anomalies.

Thus, we expect to be able to answer questions such as:

- Is the exact duration of a cache hit or miss important, or is the duration ratio miss/hit the only important parameter?
- For typical parameter values, are timing anomalies more often triggered by data cache, instruction cache or a combination thereof?
- Can one reduce or eliminate timing anomalies by choosing the right cache policy/priority?

Bibliographic Reference J. Eisinger, I. Polian, B. Becker, S. Thesing, R. Wilhelm, and A. Metzner. 2006. *Automatic Identification of Timing Anomalies for Cycle-Accurate Worst-Case Execution Time Analysis.* In *Proceedings of the 2006 IEEE Design and Diagnostics of Electronic Circuits and systems* (DDECS'06).

Benjamin Binder, Mihail Asavoae, Florian Brandner, Belgacem Ben Hedia, and Mathieu Jan. 2022. *The Role of Causality in a Formal Definition of Timing Anomalies* In *International Conference on Embedded and Real-Time Computing Systems and Applications* (RTCSA'22)

Rajesh Kumar and Mariëlle Stoelinga. 2017. *Quantitative Security and Safety Analysis with Attack-Fault Trees.* In *The 18th IEEE International Symposium on High Assurance Systems Engineering* (HASE'17)