South European Test Seminar (SETS) 2026

Using Virtual Prototypes for Causal Fault Explanations at System Level

Caroline Dominik[1] Rolf Drechsler[1,2]
[1]Institute of Computer Science, University of Bremen, Bremen, Germany
[2]Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
{cardom, drechsler}@uni-bremen.de

Abstract

Virtual Prototypes (VPs) are a useful tool in the design of complex systems. Not only do they grant developers access to internal information, which would be obscured if hardware were used instead. In addition, they allow simulating the interaction of components, with which errors that appear when joining the components can be discovered at earlier design stages. A VP-based design flow can therefore have a positive impact on the debugging process by facilitating the discovery of faults and their causes. To use this potential, we propose to include self-explanations in VPs: Based on monitoring the systems behavior during the simulation, an internal model is built, which then is used to derive causal explanations in case an error occurs.

The first challenge during this, is to avoid state space explosion when monitoring an entire system consisting of several components. This is addressed by only monitoring the information that is relevant for explaining faults at system level. For this, the variables of the software executed on the VP and the inputs and outputs of all hardware peripherals are monitored, while internals of the VP like, e.g., the registers or the timer are ignored. The product automaton of the resulting models for the software and each peripheral then describes the entire system. To further reduce the state space, an over-approximation of this automaton is used. It is created based on the execution of several benchmarks and hence only contains reachable states.

The formal model is then used to detect and explain occurring faults by applying Bounded Model Checking (BMC). BMC creates a counterexample in case of an error, which identifies an illegal trace in the system model. But it remains challenging to use it to determine the specific cause, as the load of information can be overwhelming. To derive causal explanations, we reduce the counterexample by determining the necessary condition of a failure based on the order and context of events.

The proposed explanation framework is demonstrated using an abstracted controller for a wind turbine. It consists of a low-level controller software, a wind speed sensor and a blade pitch actuator and is implemented using a RISC-V VP. Further research includes assessing and improving the scalability of the approach and the quality of the produced explanations. This includes investigating the faults and causes missed due to using an over-approximated model, as well as developing alternatives to this choice in modeling.