# Verification of Hybrid Systems Consisting of Both Complex Digital Hardware and Software

## Background

The mixture of tight time-to-market constraints and an increasing design complexity associated with the design and development of hybrid systems consisting of both digital hardware and software of today, implies that an efficient verification and qualification methodology is ever more important.

A flexible verification methodology which combines different verification and qualification methods in the most optimal way, to keep the verification and qualification cycle as short as possible while maximizing the number of issues found are key to a successful project delivery.

Verification and qualification methodologies used today, most often rely on automated regression runs and focuses highly on randomized stimuli generation. Long running regressions benefit from being executed at system level using hardware emulated systems in the lab (e.g. using FPGAs and software running on actual processors). While this approach brings the benefit of execution speed allowing a broader verification coverage, the cost of this is limited visibility during debugging of failing regressions. Verification at register-transfer-level (RTL level) on the other-hand provides high visibility into the digital system but suffers from the challenge of very low cycle speed.

Consequently, it is desirable to catch as many issues as possible during RTL development as they are hard to debug in the lab, however, the simulation speed often becomes a challenge which does not allow for verification of complete initialization sequences which may include booting of operating systems etc. hence forcing one to verify larger parts of the system at system level in the lab with reduced visibility into the system as a cost.

## Purpose

The purpose of the presented project is to obtain methods and tools that will allow increased visibility when debugging hybrid systems consisting of both hardware and software elements.

## Description

The idea is to start the development of a framework, which will consist of both methods and tools which will allow failing regressions found as part of system level verification and qualification in the lab to be moved back into a simulation environment in which the visibility into the system can be increased for better debugging.

Ideally, the same verification framework used for system level verification in the lab can be reused in simulation to allow a sub-set of the found failing regressions to be re-executed in simulation with full visibility into the system.

One idea is to establish a co-simulation framework which would allow test-cases currently executing in a Napatech proprietary test framework used for system qualification in the lab to also be executed in the co-simulation framework. The co-simulation framework could be based on QEMU for software emulation combined with RTL simulators for emulation/simulation of the digital hardware entities.

Several challenges of this approach must be addressed, e.g. regarding the ability to speed up parts of the system (e.g. in a fast-forward mode), the ability to transfer the state of the system from the emulated hardware setup in the lab into the simulation environment and vice versa and in general to tackle the notion of time and synchronization between the different modules being simulated.

It is important that the methods and framework developed can be used in a UVM based verification methodology. It would also be relevant to investigate the use of the portable stimuli specification in this context.