ulm university universität
**u**ulm

**Universität Ulm |** 89069 Ulm | Germany

**Faculty of Engineering and Computer Science**
Institute of Embedded Systems/Real-Time Systems

**Florian Hock**

Albert-Einstein-Allee 11
89081 Ulm, Germany

Tel: +49 731 50-24176
Fax: +49 731 50-24182
florian.hock@uni-ulm.de
http://www.uni-ulm.de/in/es

**2014/10/09**

**Short Term Scientific Mission**
**COST Action IC1202**

**Reference code: COST-STSM-ECOST-STSM-IC1202-060914-049971-49971**

# Scientific Report

## Purpose

In hard real-time systems, tasks need to terminate successfully until a given deadline.
Analysing the timing behaviour of tasks running on multi-core systems consisting of shared and private ressources like memories and busses requires a lot more computational effort in comparison to analysing tasks running on single-core systems with strictly private ressources.
Utilising shared ressources means, that tasks heavily interfere with each other.
In shared caches this happens by evicting useful cache blocks of a parallel running task which leads to additional cache misses of that task. Access to shared ressources by busses can lead to idle times, when access is not granted by the bus arbiter because a task running on another core also accesses the ressource at the same time.
The applicant is currently working on a code-level optimisation which utilises private Scratch-Pad memories (SPM) to reduce the idle times of parallel executed tasks executed on a multi-core system with a TDMA arbitrated system bus. This optimisation integrates the timing-analysis of the system bus into the ILP-formulated allocation of code fragments onto the SPM.
Integrating timing and interference analysis into an ILP-formulated optimisation increases the computational effort for solving the optimisation problem. Especially, when considering cache analysis and bus arbitration schemes others than TDMA, like priority based ones, which is planned by the applicant, the computational effort can be driven to the edge of infeasibility.

## Activities during the STM

The STSM tool place from Sept. 08th, until Sept. 14th, 2014.

On Monday Sept. 08th, 2014 a first meeting of the NUS group and the Uni Ulm group was set up. Overviews about the current and future scientific work of both groups where given in several presentations. After that, first discussions took place to find starting-points for future cooperative work.
Tuesday until Friday, ideas and concepts were worked out to combine the work for test generation of the NUS group and the applicants group.
A wrap-up meeting on Friday Sept. 12th, was held to finalise the ideas prepared during the week.

## Main Results

The NUS group provides a framework [1] for generating test cases for worst-case timing behaviours and utilising repairing techniques subsequently. It would be integrated into the WCET-aware C-Compiler (WCC) [2] of the Uni Ulm group which provides the ability to integrate analysing and optimising techniques.
The framework of the NUS group uses static analysis as a first stage, which leads the dynamic test generation via an interface which systematically generates assertions to find performance stressing test inputs.
Currently the analysis and the interface for assertion generation covers the caching behaviour of a program. This can be modified to cover the behaviour of other micro-architectural components such as bus arbiters.
Focusing on scenarios which lead to bad timing behaviour in the optimisation process will reduce the computational complexity.

## Future Collaboration

Both groups agreed to share their infrustructure to allow the mutual integration of tools for future research.

## Publications

As the framework of the NUS group helps to reduce the computational complexity of optimisation problems it will have a major impact on the future publications of the applicant.

## Bibliography

[1] Abhijeet Banerjee, Sudipta Chattopadhyay and Abhik Roychoudhury, “Static Analysis driven Cache Performance Testing”,*In Proceedings of RTSS*. 2013

[2] H. Falk and P. Lokuciejewski. “A compiler framework for the reduction of worst-case execution times”, *Springer Journal of Time-Critical Computing Systems (Real-Time Systems)*, 46(2):251–300, 2010.