# inf334 - System Level Design

<table>
<thead>
<tr>
<th>Module label</th>
<th>System Level Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>inf334</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
<tr>
<td>Used in course of study</td>
<td></td>
</tr>
<tr>
<td>Master's Programme Computing Science &gt; Technische Informatik</td>
<td></td>
</tr>
<tr>
<td>Master's Programme Engineering of Socio-Technical Systems &gt; Embedded Brain Computer Interaction</td>
<td></td>
</tr>
<tr>
<td>Master's Programme Engineering of Socio-Technical Systems &gt; Systems Engineering</td>
<td></td>
</tr>
</tbody>
</table>

## Contact person

**Module responsibility**
- Kim Grüttner

**Authorized examiners**
- Die im Modul Lehrenden
- Kim Grüttner

## Entry requirements

**Skills to be acquired in this module**

**Professional competences:**
- The students:
  - Ability to describe and analyze system components and architectures using system level description languages SpecC and SystemC
  - Capabilities for partitioning and parallelizing of applications

**Methodological competences:**
- The students:
  - Knowledge of refinement and transformation techniques for transferring an initial specification into a real implementation
  - Knowledge of the phases of a system-level design flow
  - Knowledge of current design methods and tools in system level design
  - Knowledge about formal models of computation of specification languages
  - Knowledge of current research results and trends in system level design
  - Capabilities for partitioning and parallelizing of applications
  - Ability to evaluate and explore design decisions
  - Ability to implement a complete system design-to-implementation specification

**Social competences:**
- The students:
  - Implement solutions of given problems in teams
  - Discuss their outcomes appropriately

**Self-competences:**
- The students:
  - presentation skills
  - reflect their solutions by using methods learned in this course

## Module contents

The ever-increasing integration densities of integrated circuits enable the implementation of increasingly powerful and complex systems. This can be on the one hand the integration of several sub-components on the same chip (system-on-chip) or on the other hand the implementation of more powerful algorithms. However, traditional design techniques are hardly able to cope with the increasing complexity of today's embedded systems. Therefore, in research and practice efforts through new methods and tools, there is a significant increase in productivity in the design process, thus closing the so-called "design productivity gap". This is achieved, for example, by a stronger abstraction, in which the behavior of components is described only at the algorithmic level and is automatically translated into hardware or software implementations by high-level synthesis techniques. The final system implementation is achieved by means of a structured refinement and exploration processes. Throughout this refinement flow, system properties (for example, timing,
energy consumption, chip area and costs) are estimated on each abstraction level and guide the
designer in the iterative decision process. By means of techniques such as virtual prototyping, entire
systems can be simulated and verified on each refinement layer, even without the availability of a full
implementation for all system components.

This module builds on the modules Embedded Systems I and II, deepens the knowledge acquired
there for the design of hardware/software systems and expands them with current methods and
tools. With SystemC, a language is presented that is already widely used in industry and research for
the design and verification of hardware/software systems and supports several abstraction levels
from clock cycle accurate hardware description, over transaction level models to process based
functional specifications.

Reader's advisory

Suggested reading:

Main textbooks:

- D. Gajski, S. Abdi, A. Gerstlauer, G. Schirner, Embedded System Design: Modeling,
- D. C. Black, J. Donovan, B. Bunton, A. Keist, SystemC: From The Ground Up, Second

Optional books:

- F. Vahid, T. Givargis, Embedded System Design: A Unified Hardware/Software Introduction,
  Wiley, John & Sons, 2001 (“blue book”), Background about embedded systems in general

Additional reading material posted on Stud.IP

Links

https://www.uni-oldenburg.de/informatik/ehs/lehre/vorlesungen/system-level-design/

Language of instruction
English

Duration (semesters)
1 Semester

Module frequency
once a year

Module capacity
unlimited

Module level
AS (Akzentsetzung / Accentuation)

Modulart
Pflicht o. Wahlpflicht / compulsory or optional

Vorkenntnisse / Previous knowledge
- inf200 Grundlagen der Technische Informatik,
- inf201 Technische Informatik,
- inf203 Eingebettete Systeme I,
- inf204 Eingebettete Systeme II

Final exam of module

Time of examination
at the end of the lecture period

Type of examination
hands-on exercises and oral exam

Course type
Comment
SWS
Frequency
Workload attendance
Lecture
2.00
SuSe
28 h
Exercises
2.00
SuSe
28 h

Total time of attendance for the module
56 h