Linköping University
Embedded Systems Laboratory

www.ida.liu.se/~eslab

Call for Participation in a SYDIC-Training Course on

System-Level Design of Embedded Systems

Course Contents

Part I: Lectures

Lecture 1: Introduction (2 hours)
- Embedded systems and their design
- Characteristics and requirements

Lecture 2: Models of computation and specification languages (5 hours)
- System specification and formal methods
- Models of computation
- Dataflow models
- Petri nets
- Discrete event models
- Synchronous finite state machines & synchronous languages
- Codesign finite state machines & the POLIS system
- Timed automata
- What modeling approach and specification language to choose?

Lecture 3: Embedded system architectures (3 hours)
- Component and platform-based design
- Architecture specialization techniques
- Typical architectures for embedded systems
- Design space exploration

Lecture 4: Real-time embedded systems: task scheduling (3 hours)
- Real-time systems and their typical features
- Worst-case execution time analysis
- Task scheduling policies
- Static cyclic scheduling
- Priority based scheduling
- Schedulability analysis

Objective

The aim of the course is to understand the particular problems concerning the design of complex embedded systems. To learn about modern design methodologies with an emphasis on early design phases, not covered by traditional methods, including modeling, verification and system-level synthesis. It also discusses the topics of embedded real-time systems, and power related issues at the system level.

Target Audience

Designers and engineers interested in system-level design and modeling for embedded systems.

Duration

3 days.

Date

May 5-7, 2004.

Venue

Linköping University, Sweden.

Organizer

Prof. Zebo Peng.

Lecture 5: System-level power/energy optimization (3 hours)

- Sources of power dissipation
- System level power optimization
- Dynamic power management
- Mapping and scheduling for low energy
- Real-time scheduling with dynamic voltage scaling

Part II: Lab exercise (8 hours)

The lab exercises will illustrate the use of modeling techniques and the associated formal verification tools. The focus is on model checking, one of the well-established automatic approaches in formal methods. Model checking is used to determine whether the model of a system satisfies a set of required properties. The software tools to be used include the model checkers SMV and Uppaal.

Exercise 1: Model Checking of Finite State Systems

The first exercise shows how to describe finite state systems using the input language of the tool SMV and also how to formalize required properties in the temporal logic CTL (Computation Tree Logic). Having as input the system description and the desired properties in a formal notation, the model checker answers whether the model of the system satisfies the given properties. If not, counter-examples are generated that can be further used as diagnosis information.

Exercise 2: Model Checking of Continuous-Time Systems

The second exercise illustrates how to represent, validate, and verify real-time systems based on timed automata. A timed automaton is a finite automaton enhanced with a set of real-valued clocks that may model systems where time advances continuously. Uppaal is a model checker that allows to graphically specify continuous-time systems and to perform their validation/verification. The logic for specifying properties is TCTL (Timed CTL), a real-time extension of CTL that allows to inscribe subscripts on the temporal operators to limit their scope in time.

The course is given in the framework of the EU project IST-2001-35100 SYDIC-Training. Additional information can be found at

www.ecsi.org/training

Registration and contact

Prof. Zebo Peng
Embedded Systems Laboratory
Dept. of Computer and Information Science
Linköping University

SE-581 83 Linköping
Sweden

Phone: +46 13 28 20 67
Fax: +46 13 14 22 31
E-mail: zpe@ida.liu.se

www.ida.liu.se/~eslab