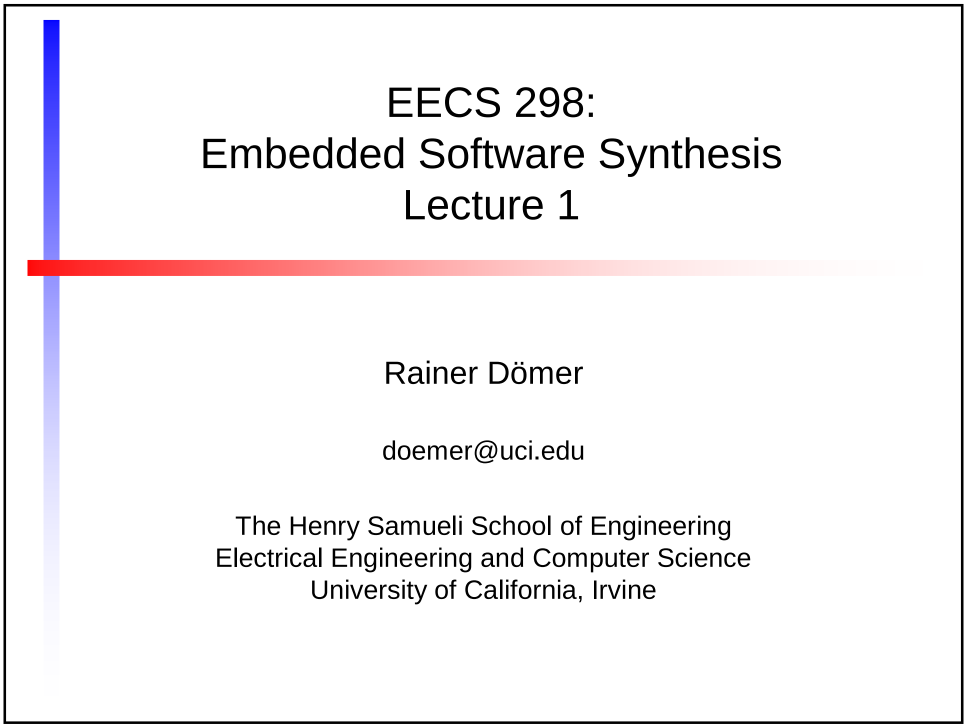
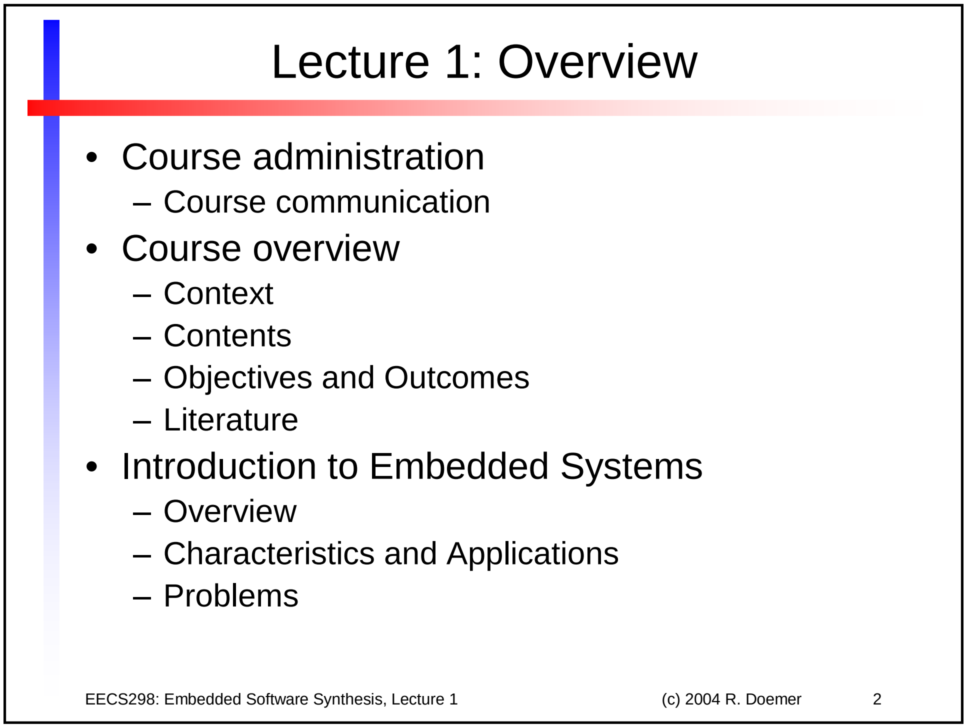


EECS 298: Embedded Software Synthesis Lecture 1

Rainer Dömer

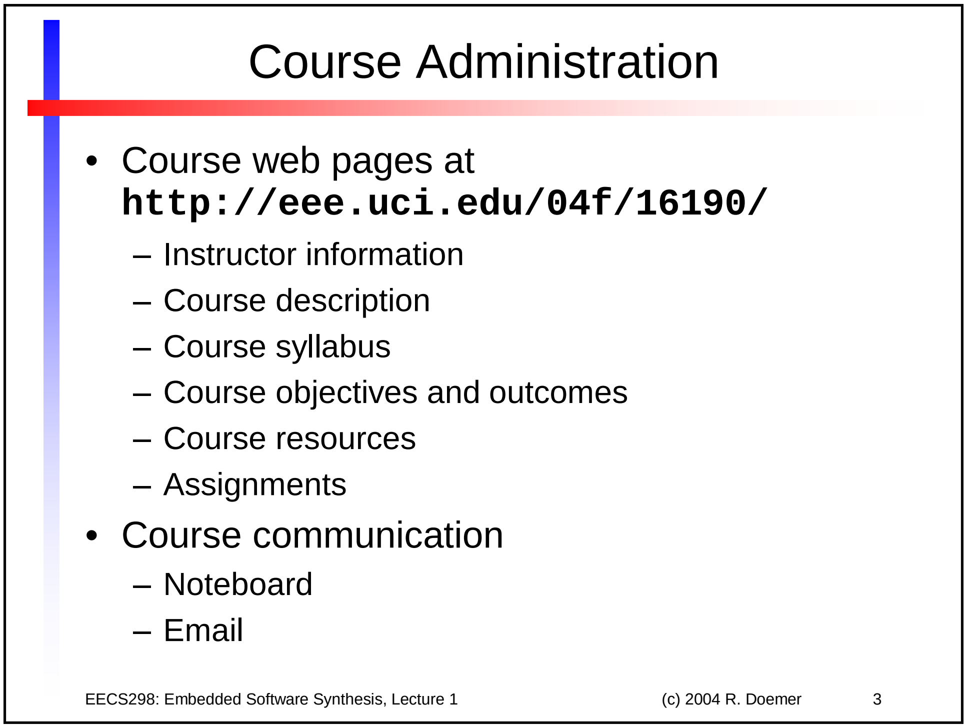
doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine



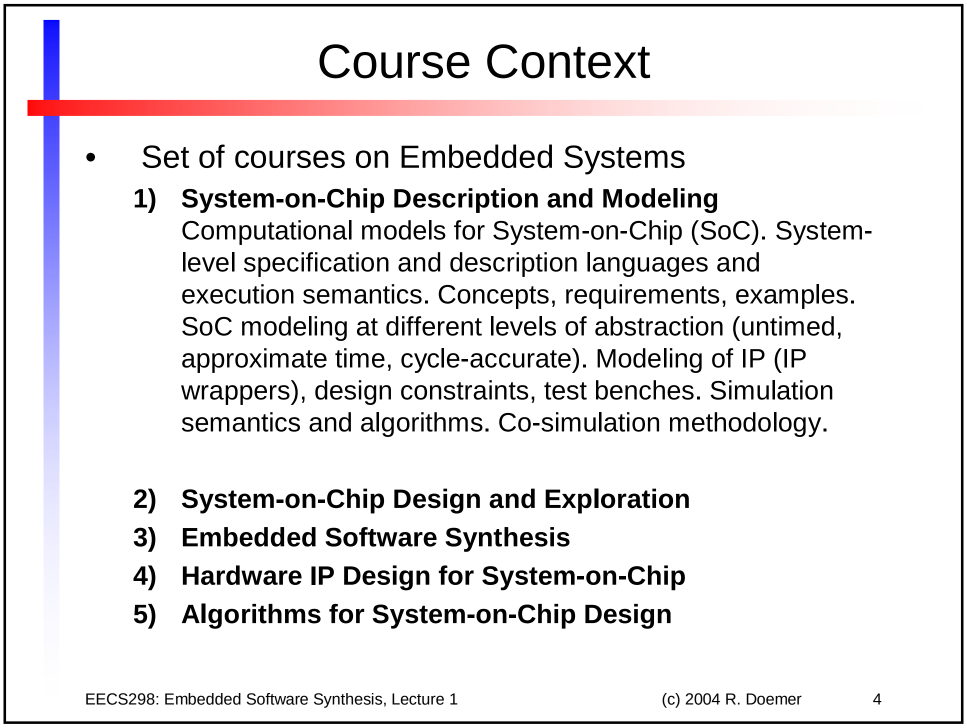
Lecture 1: Overview

- Course administration
  - Course communication
- Course overview
  - Context
  - Contents
  - Objectives and Outcomes
  - Literature
- Introduction to Embedded Systems
  - Overview
  - Characteristics and Applications
  - Problems



Course Administration

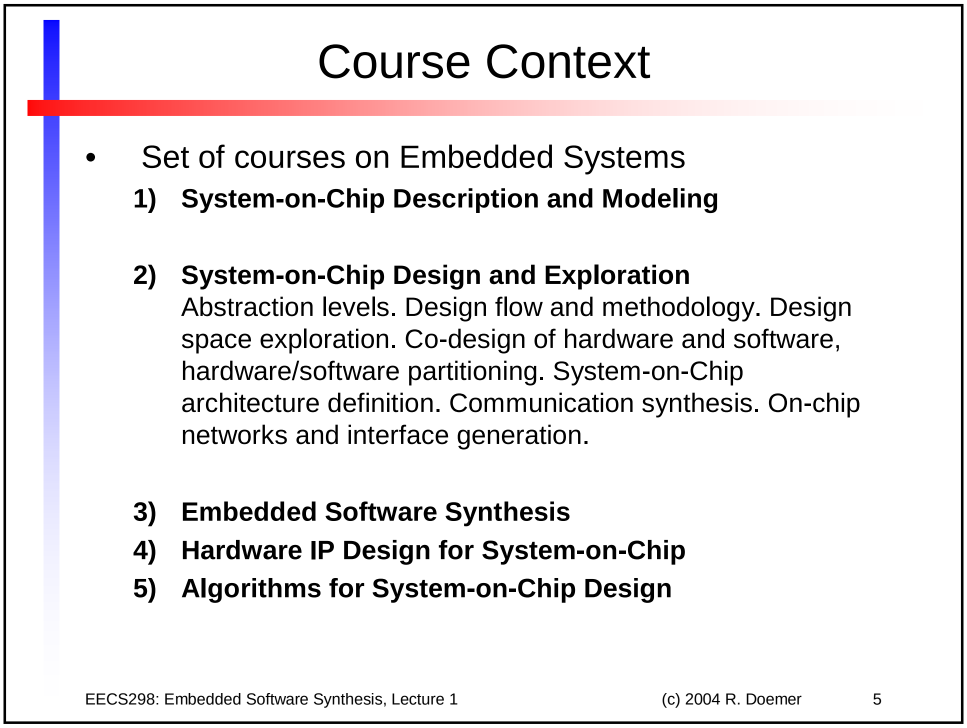
- Course web pages at **`http://eee.uci.edu/04f/16190/`**
  – Instructor information
  – Course description
  – Course syllabus
  – Course objectives and outcomes
  – Course resources
  – Assignments
- Course communication
  – Noteboard
  – Email



Course Context

- Set of courses on Embedded Systems
  1) **System-on-Chip Description and Modeling**
  Computational models for System-on-Chip (SoC). System-level specification and description languages and execution semantics. Concepts, requirements, examples. SoC modeling at different levels of abstraction (untimed, approximate time, cycle-accurate). Modeling of IP (IP wrappers), design constraints, test benches. Simulation semantics and algorithms. Co-simulation methodology.

  2) **System-on-Chip Design and Exploration**
  3) **Embedded Software Synthesis**
  4) **Hardware IP Design for System-on-Chip**
  5) **Algorithms for System-on-Chip Design**



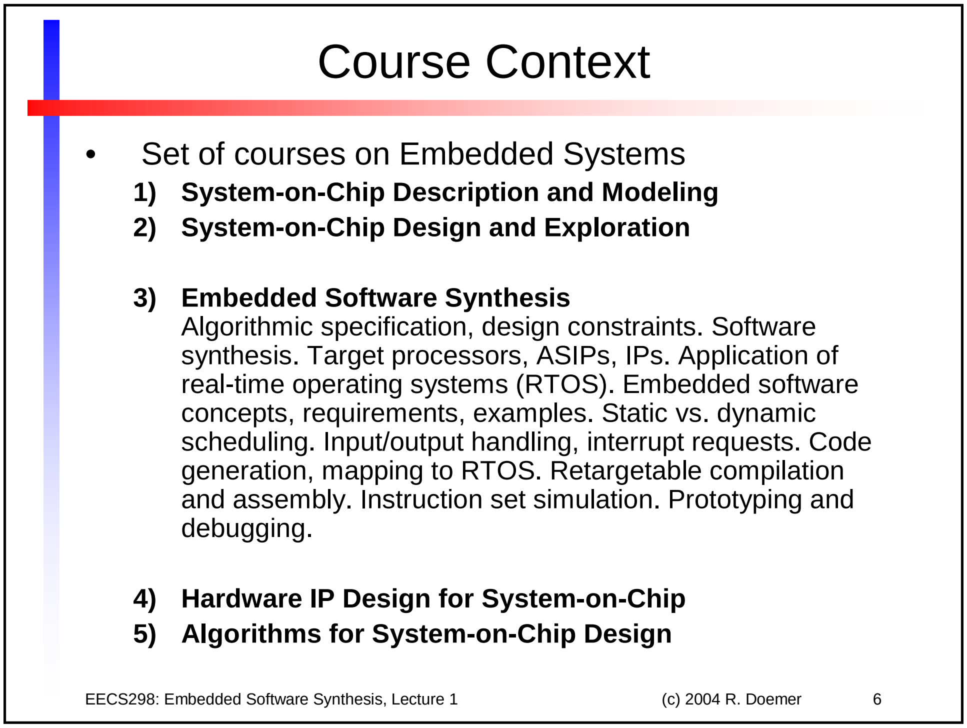
Course Context

- Set of courses on Embedded Systems

1) **System-on-Chip Description and Modeling**

2) **System-on-Chip Design and Exploration**
   Abstraction levels. Design flow and methodology. Design space exploration. Co-design of hardware and software, hardware/software partitioning. System-on-Chip architecture definition. Communication synthesis. On-chip networks and interface generation.

3) **Embedded Software Synthesis**
4) **Hardware IP Design for System-on-Chip**
5) **Algorithms for System-on-Chip Design**



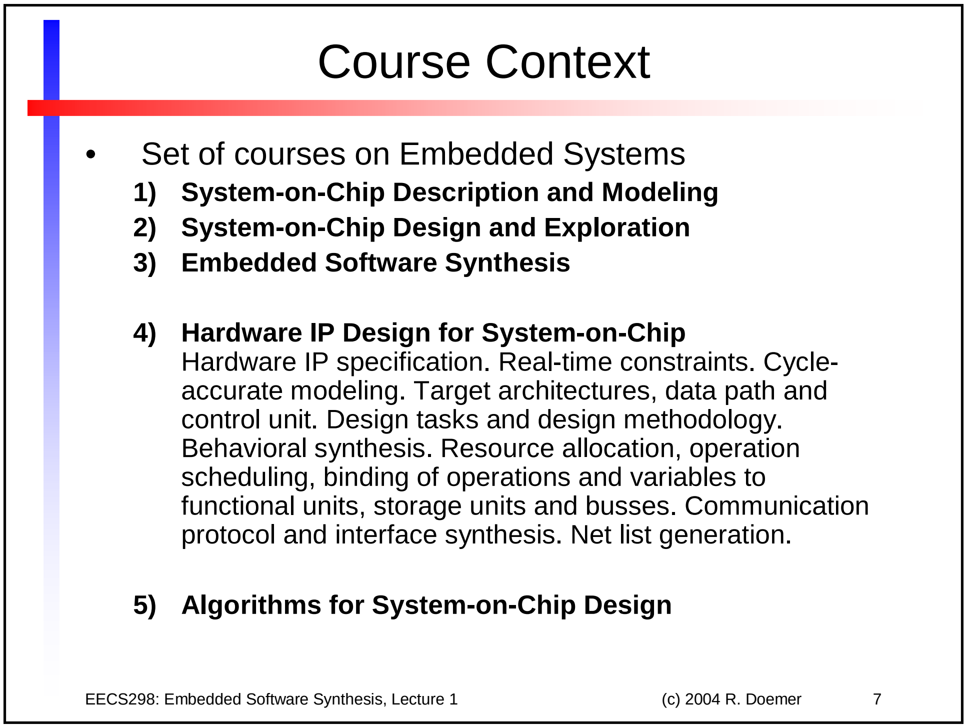
Course Context

- Set of courses on Embedded Systems

1) **System-on-Chip Description and Modeling**
2) **System-on-Chip Design and Exploration**

3) **Embedded Software Synthesis**
   Algorithmic specification, design constraints. Software synthesis. Target processors, ASIPs, IPs. Application of real-time operating systems (RTOS). Embedded software concepts, requirements, examples. Static vs. dynamic scheduling. Input/output handling, interrupt requests. Code generation, mapping to RTOS. Retargetable compilation and assembly. Instruction set simulation. Prototyping and debugging.

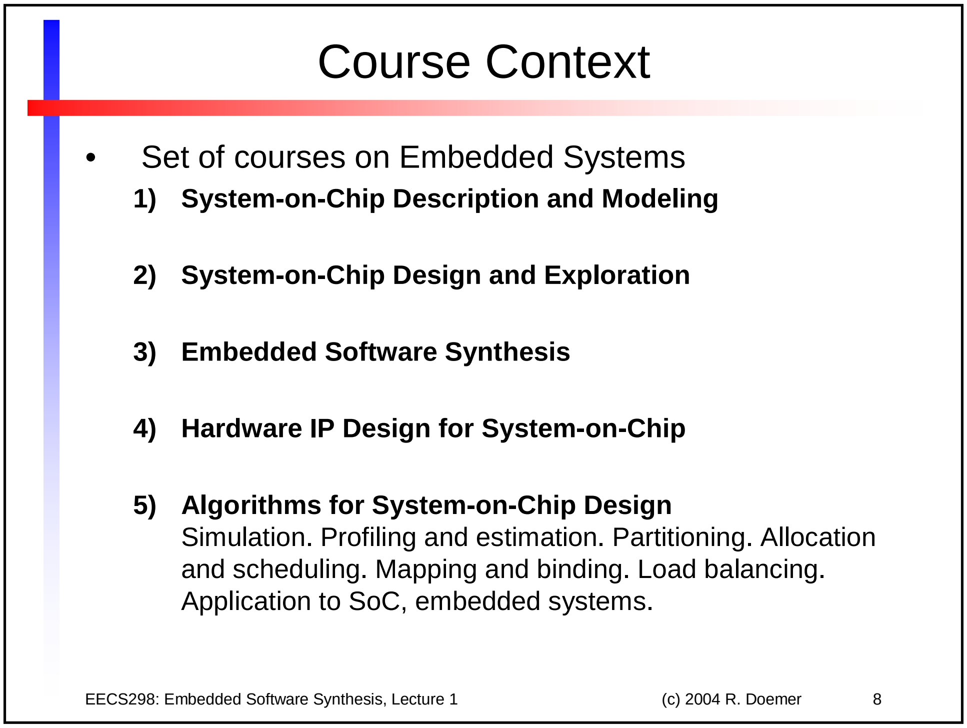
4) **Hardware IP Design for System-on-Chip**
5) **Algorithms for System-on-Chip Design**



Course Context

- Set of courses on Embedded Systems

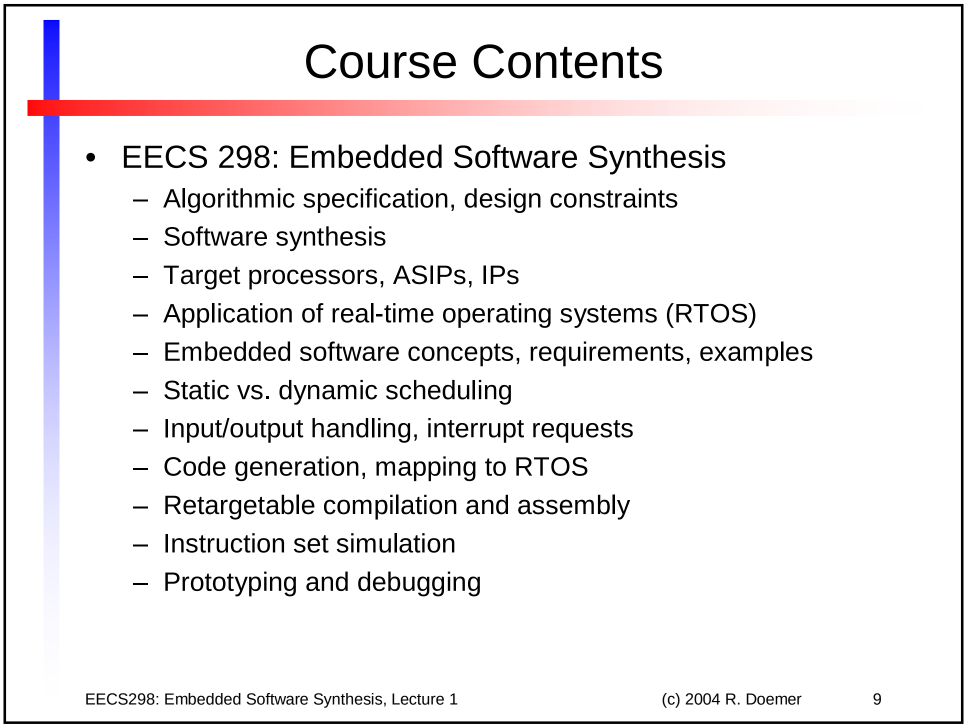
1) **System-on-Chip Description and Modeling**
2) **System-on-Chip Design and Exploration**
3) **Embedded Software Synthesis**
4) **Hardware IP Design for System-on-Chip**
   Hardware IP specification. Real-time constraints. Cycle-accurate modeling. Target architectures, data path and control unit. Design tasks and design methodology. Behavioral synthesis. Resource allocation, operation scheduling, binding of operations and variables to functional units, storage units and busses. Communication protocol and interface synthesis. Net list generation.
5) **Algorithms for System-on-Chip Design**



Course Context

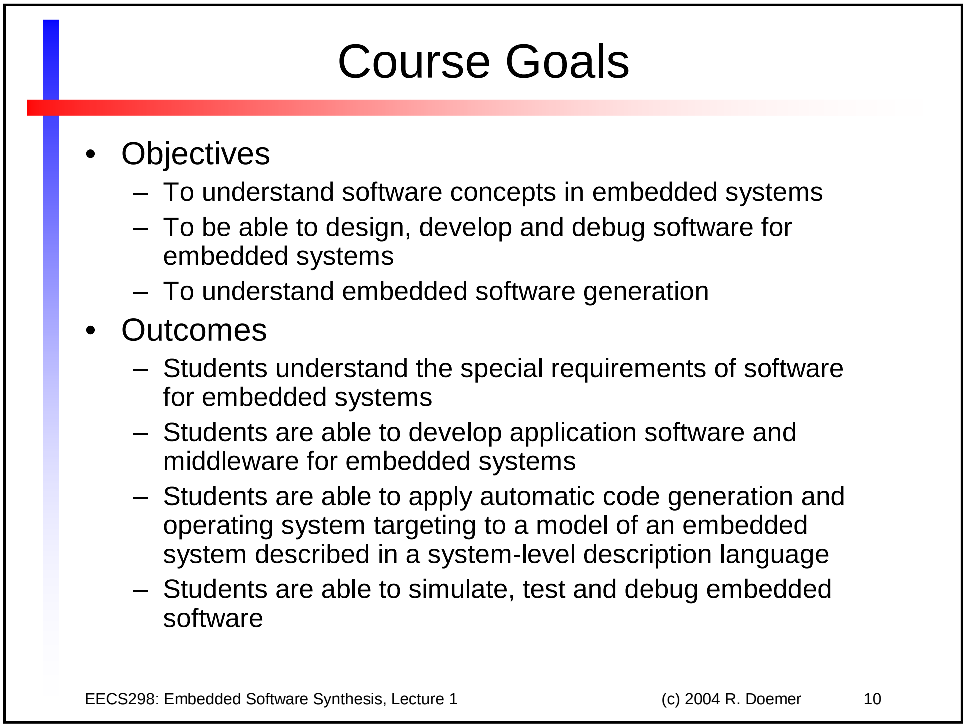
- Set of courses on Embedded Systems

1) **System-on-Chip Description and Modeling**
2) **System-on-Chip Design and Exploration**
3) **Embedded Software Synthesis**
4) **Hardware IP Design for System-on-Chip**
5) **Algorithms for System-on-Chip Design**
   Simulation. Profiling and estimation. Partitioning. Allocation and scheduling. Mapping and binding. Load balancing. Application to SoC, embedded systems.



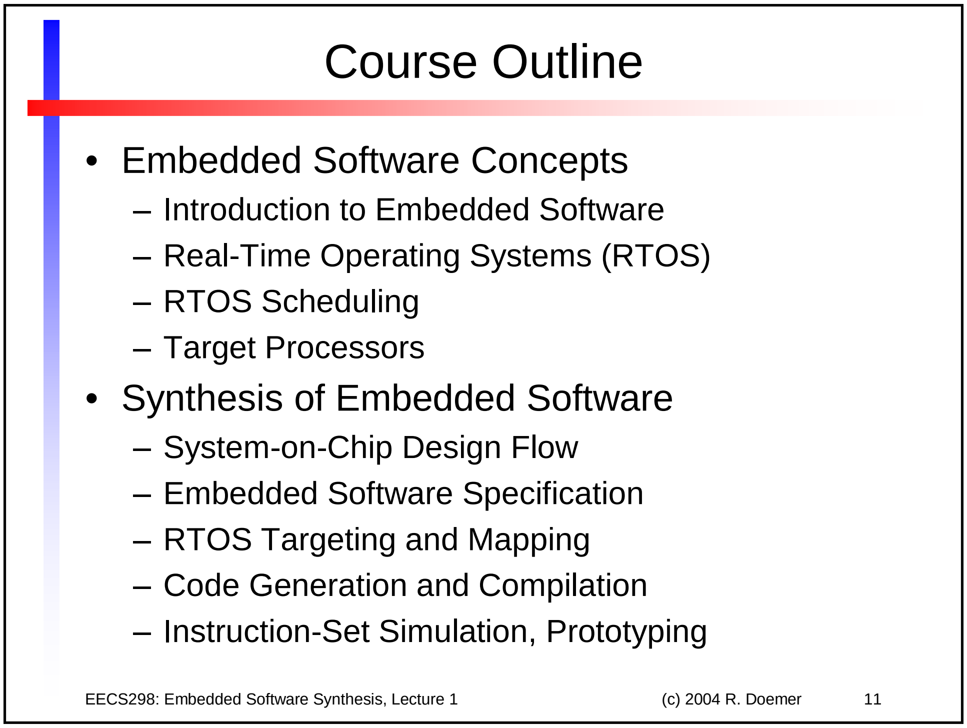
Course Contents

- EECS 298: Embedded Software Synthesis
  - Algorithmic specification, design constraints
  - Software synthesis
  - Target processors, ASIPs, IPs
  - Application of real-time operating systems (RTOS)
  - Embedded software concepts, requirements, examples
  - Static vs. dynamic scheduling
  - Input/output handling, interrupt requests
  - Code generation, mapping to RTOS
  - Retargetable compilation and assembly
  - Instruction set simulation
  - Prototyping and debugging



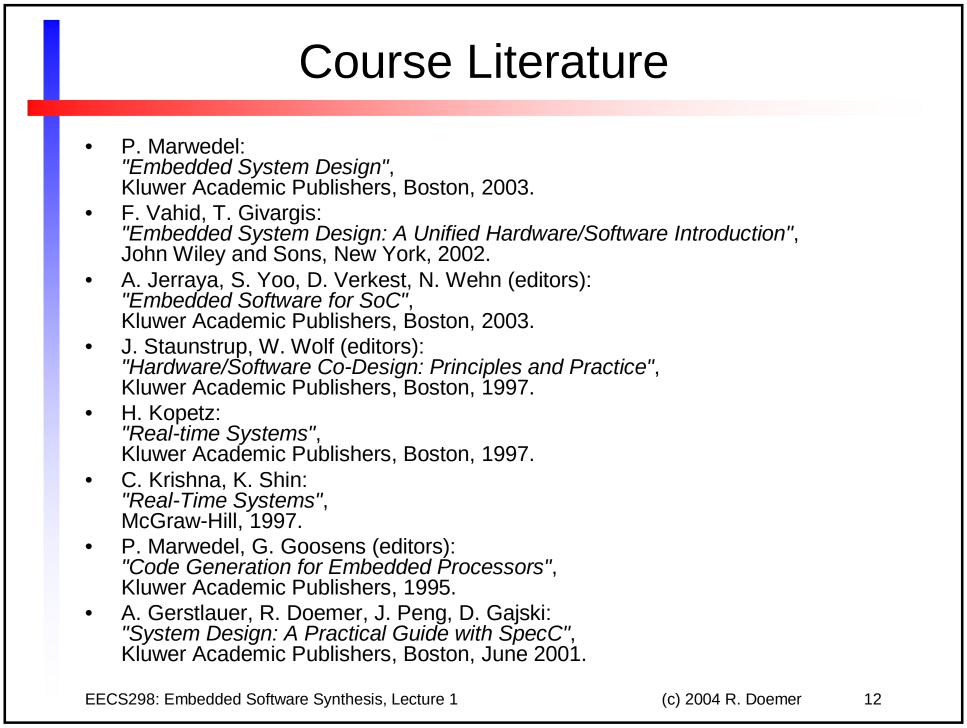
Course Goals

- Objectives
  - To understand software concepts in embedded systems
  - To be able to design, develop and debug software for embedded systems
  - To understand embedded software generation
- Outcomes
  - Students understand the special requirements of software for embedded systems
  - Students are able to develop application software and middleware for embedded systems
  - Students are able to apply automatic code generation and operating system targeting to a model of an embedded system described in a system-level description language
  - Students are able to simulate, test and debug embedded software



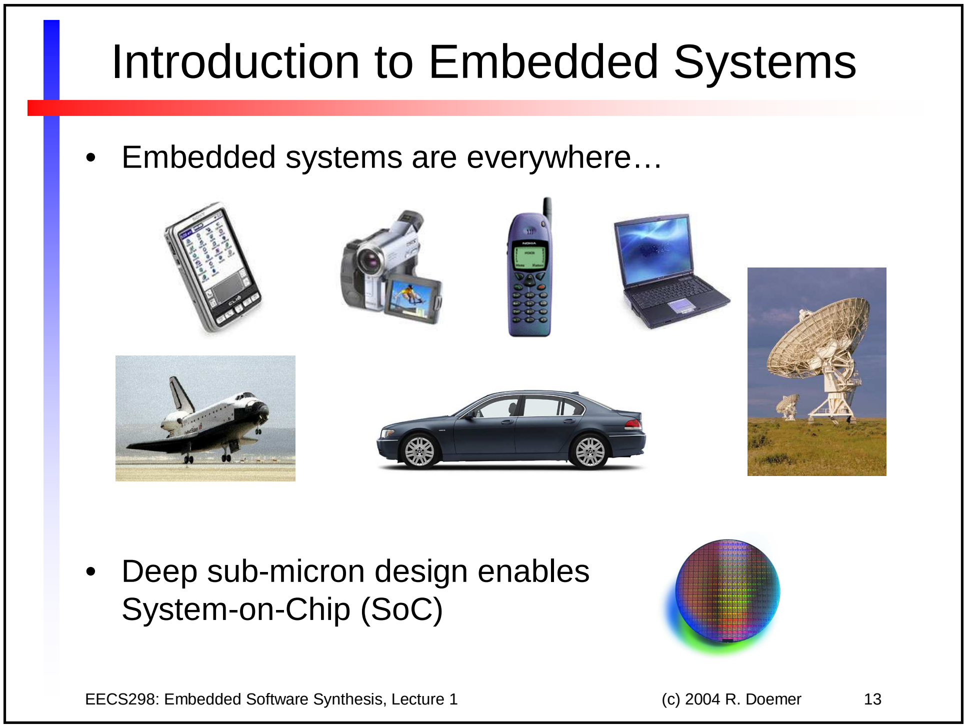
Course Outline

- Embedded Software Concepts
  - Introduction to Embedded Software
  - Real-Time Operating Systems (RTOS)
  - RTOS Scheduling
  - Target Processors
- Synthesis of Embedded Software
  - System-on-Chip Design Flow
  - Embedded Software Specification
  - RTOS Targeting and Mapping
  - Code Generation and Compilation
  - Instruction-Set Simulation, Prototyping



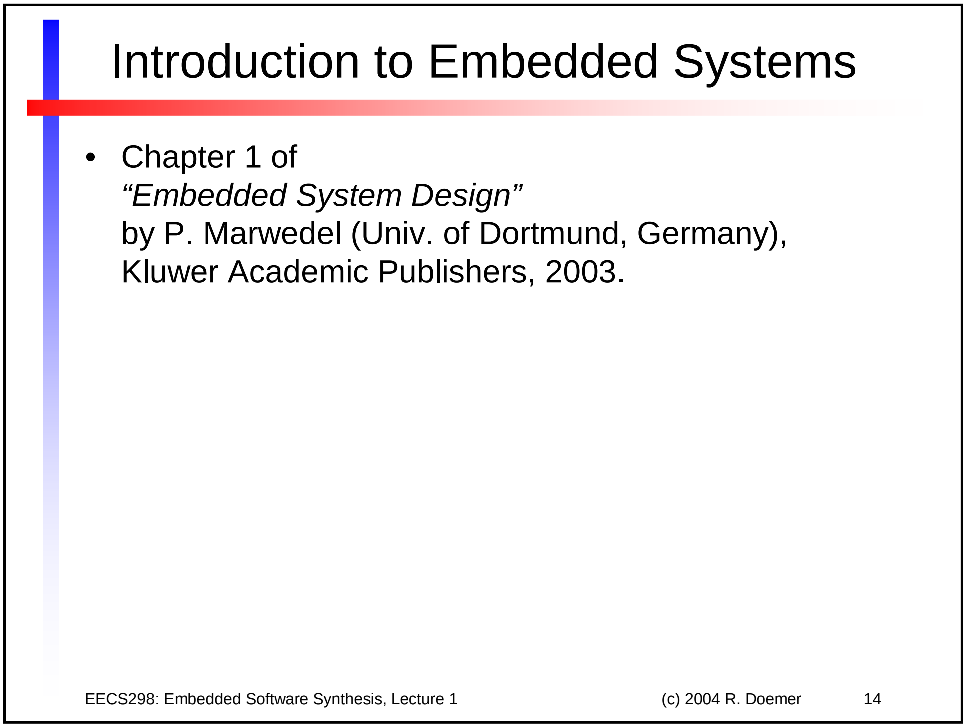
Course Literature

- P. Marwedel:
  "Embedded System Design",
  Kluwer Academic Publishers, Boston, 2003.
- F. Vahid, T. Givargis:
  "Embedded System Design: A Unified Hardware/Software Introduction",
  John Wiley and Sons, New York, 2002.
- A. Jerraya, S. Yoo, D. Verkest, N. Wehn (editors):
  "Embedded Software for SoC",
  Kluwer Academic Publishers, Boston, 2003.
- J. Staunstrup, W. Wolf (editors):
  "Hardware/Software Co-Design: Principles and Practice",
  Kluwer Academic Publishers, Boston, 1997.
- H. Kopetz:
  "Real-time Systems",
  Kluwer Academic Publishers, Boston, 1997.
- C. Krishna, K. Shin:
  "Real-Time Systems",
  McGraw-Hill, 1997.
- P. Marwedel, G. Goosens (editors):
  "Code Generation for Embedded Processors",
  Kluwer Academic Publishers, 1995.
- A. Gerstlauer, R. Doemer, J. Peng, D. Gajski:
  "System Design: A Practical Guide with SpecC",
  Kluwer Academic Publishers, Boston, June 2001.



Introduction to Embedded Systems

- Embedded systems are everywhere…
- Deep sub-micron design enables System-on-Chip (SoC)



Introduction to Embedded Systems

- Chapter 1 of
  “Embedded System Design”
  by P. Marwedel (Univ. of Dortmund, Germany),
  Kluwer Academic Publishers, 2003.