

ESE 566: Hardware/Software Co-Design of Embedded Systems

Fall 2004

Instructor: Alex Doboli.

Credits: 3 credits

Description: This course will present state-of-the-art concepts and techniques for design of embedded systems consisting of hardware and software components. Discussed topics include system modeling and specification, architectures for embedded systems, performance evaluation, and synthesis. The course follows the top-down design paradigm based on IP cores. Course requirements include four mini-projects focused on system and IP core specification, and implementation of various co-design tools.

Goal: Upon completion of this course, students will possess knowledge on state-of-the-art methodologies and techniques for hardware/software co-design of embedded systems. They will be able to (1) develop system-level specifications using high-level languages i.e. SystemC, (2) develop an IP core models in SystemC, and (3) implement algorithms for co-design.

Text Book and other Teaching Material:

1. G. De Micheli, R. Ernst, W. Wolf, "Readings in Hardware/Software Co-Design", *Morgan Kaufman*, 2002.
2. G. De Micheli, "Synthesis and Optimization of Digital Circuits", *McGraw-Hill*, 1994.
3. Other published papers will be provided in class.

Prerequisites: ESE 545 (Computer Architectures), ESE 554 (Computational Models), and ESE 333 (Real-Time Operating Systems) or equivalent.

Covered Topics:

- 1) **Introduction to Co-Design:**
 - a. Problem description, goals of co-design, co-design steps, existing co-design approaches, and present challenges.
- 2) **System Modeling and Specification:**
 - a. Models of computation (Data flow model, Task graphs, Petri nets, Finite State Machines, Extended Finite State Machines, hierarchical models etc).
 - b. System specification languages (SystemC).
 - c. *Modeling of an application in SystemC (project 1).*
- 3) **Architectures for Embedded Systems:**
 - a. Single processor – coprocessor architecture, multiprocessor architectures, architectures for DSP and multimedia, reconfigurable architectures, platforms.
 - b. SoC: IP core based architectures, bus standards, networks on chip.
 - c. Sensors and sensor networks (**tentative**)
 - d. *IP core modeling in System C (project 2).*
- 4) **Performance Modeling:**
 - a. System-level performance modeling vs. low-level performance modeling.
 - b. Modeling of system latency, energy consumption etc for hardware and software.

- c. Estimation of memory requirements.
 - d. Stochastic modeling (**tentative**).
- 5) **System-Level Synthesis and Trade-off Analysis:**
- a. Hardware/software partitioning. Task binding.
 - b. Task scheduling (scheduling under data and control dependencies, static and dynamic scheduling, heuristic and exact scheduling algorithms).
 - c. IP core integration and communication synthesis: Hardware and software interface synthesis, bus encoding for low power, bus architecture synthesis.
 - d. *Projects 3 and 4.*
- 6) **Hardware IP core synthesis:**
- a. High-level synthesis: behavioral specification of hardware, module set allocation, resource binding, operation scheduling, controller design.
- 7) **Software synthesis (tentative):**
- a. Embedded software design. Software generation under memory and energy consumption minimization constraints.

Other Course Material:

- 1) R. Ernst, "Codesign of Embedded Systems: Status and Trends", *IEEE Design & Test of Computers*, June 1998.
- 2) K. Keutzer et al, "System-level design: orthogonalization of concerns and platform-based design", *IEEE Transaction on CADICS*, December 2000.
- 3) E. Lee et al, "A Framework for Comparing Models of Computation ", *IEEE Transactions on CADICS*, December 1998, pp. 1217-1229.
- 4) R. Ernst et al, "Hardware-Software Co-synthesis for Microcontrollers", *IEEE Design & Test of Computers*, December 1993.
- 5) R. Gupta et al, "A Cosynthesis Approach to Embedded System Design Automation", *Journal of Design Automation of Embedded Systems*, January 1996.
- 6) W. Wolf, "An Architectural Co-synthesis Algorithm for Distributed, Embedded Computing Systems", June 1997.
- 7) J. Henkel et al, "A Hardware-Software Partitioning Using a Dynamically Determined Granularity", *Proc. Design Automation Conference*, 1997.
- 8) Balarin et al, "Synthesis of software programs for embedded control applications", *IEEE Transactions on CADICS*, June 1999, pp. 834-849.
- 9) P. Eles et al, "Scheduling with Bus Access Optimization for Distributed Embedded Systems", *IEEE Transactions on VLSI Systems*, October 2000.
- 10) P. Chou et al, "ipChinook: An integrated IP-based Design Framework for Distributed Embedded Systems", *Proc. Design Automation Conference*, 1999.
- 11) Y. Li, S. Malik, "Performance Analysis of Embedded software using implicit path enumeration", *IEEE Transactions on CADICS*, December 1997.
- 12) Other relevant papers.

Grading:

$$\text{Final grade} = 0.4 * \text{Final project} + 0.2 * \text{Project1} + 0.2 * \text{Project2} + 0.2 * \text{Project3}$$