EE554 Cyber-Physical Systems: A Computing Perspective

Units: 4
Fall 2024
Lecture: Mon & Wed: 8:00am-9:50am (lecture time)
Mon & Wed, 10:00am - 11:00am (office hours)
Discussion: Tue: 9:00am-10:00am
Laboratory: Fri: 9:00am – 10:50am

Location: ZHS 352
Website: USC Blackboard

Instructor: Paul Bogdan
Office: EEB 304 (Hughes Aircraft Electrical Engineering Building)
Office Hours: Mon & Wed 10:00am - 11:00am (office & online)
Contact Info: Email: pbogdan@usc.edu Phone: 213-821-5720
I am replying to emails as soon as I read them (within hours).

Teaching Assistant: Shukai Duan
Office: EEB 351
Office Hours: Tue & Thu, 6:00pm-7:00pm
Contact Info: shukaidu@usc.edu

Course Description

The laws of physics dictated not only the technological advances, but also the observed trends in embedded computing from single- to many-core systems. We are witnessing a manycore percolation in all domains from Cyber-Physical Systems (CPS) (e.g., including wearable, portable and medical devices in healthcare, trustworthy perception and decision making computing modules in unmanned aerial / ground / underwater vehicles) to intelligent Internet-of-Things (IoT) for controlling the transportation infrastructure, semi- and fully-autonomous vehicles, trains and avionics, to embedding proactive decision making in the power grid and energy systems. Workloads ranging from high performance simulation (for analysis and forecasting tasks) and complex bio-physico-chemistry modeling to data analytics and deep learning algorithms are all expected to benefit from highly parallel extremely heterogeneous manycore systems. However, marching towards exascale computing systems is not an easy task and is poised by a number of theoretical and technological challenges concerning the need for expressive and scalable models of computation describing both physical and cyber components, compact an accurate workload / application models, rigorous and efficient application to architecture compiler and performance analysis tools, efficient design and optimization algorithms of application software, intelligent distributed operating system and heterogeneous hardware.

In this course, we will provide: (1) a comprehensive overview of mathematical models for CPS, (2) new mathematical concepts (e.g., graph multifractal spectrum, graph curvature) and algorithmic tools for mining the complexity of high level programs and discovering in real-time the optimal degree of parallelization for an incoming set of interdependent applications in the context of stream and steering processing, (3) new intelligent compiler, mapping and scheduling algorithms for determining the type and number of processing elements, the amount and hierarchy of heterogeneous memory and the required reconfigurable on- and off-chip interconnect for meeting the real time requirements of CPS, (4) new algorithms for high-level synthesis embedded in more efficient design methodologies for distributed manycore-based CPS, (5) overview of existing tools and recent mathematical strategies and algorithms for formal performance analysis, verification, testing and validation of AI-based autonomous CPS. A special emphasis will be put on understanding the mathematical models required to describe the characteristics of each domain application, express features and constraints and translate into primitives for efficient design of manycores that make up the CPS component in various application domains (transportation, avionics, healthcare, space exploration). To acquire deep knowledge in manycore-based CPS design, a semester long project will focus on tackling all issues from specification, to modeling, to performance analysis and optimization, to prototyping, validation, verification and testing.

Learning Objectives

The learning objectives of this course are:

- To apply mathematical definitions and concepts to the design of new models of computation for future applications in domains such as healthcare, transportation, and avionics;
- To generalize and develop new algorithms for specific CPS design problems (e.g., identifying strategies for parallelization, estimating the timing characteristics of applications on specific hardware architectures, exploiting mathematical characteristics of CPS high-level programs for determining the best type of processing elements, memory and interconnect) and CPS optimization in response to developments in software and hardware;
- To assess the accuracy of profiling of high-level language (HLL) programs and their implications for computation, communication and memory transactions;
- To formulate and solve system-level optimization problems while considering performance, power, reliability as cost functions with implications for CPS design methodologies;
- To apply theoretical concepts to concrete problems and identify the optimal design of a heterogeneous manycore platform for specific application domains and codes (i.e., starting from an English or a high-level program description of a CPS application and a set of CPS resources, determine the optimal CPS architecture with respect to real-time requirements / performance, power / energy consumption, or reliability);
- To evaluate a proposed solution both analytically and experimentally (e.g., through computational thinking and simulation environments, FPGA prototyping)
- To design, synthesize, and assess the proposed CPS design against existing architectures, as well as to summarize the benefits and limitations of the prototyped CPS design against the proposed analytical solution
- To communicate research results and findings in a logical and persuasive manner

The lectures, project milestones, laboratory and homework assignments are progressively following these learning objectives.

Required Readings and Supplementary Materials

There is no single textbook containing all subjects to be discussed, reviewed and developed in this class. During each lecture, a number of scholarly articles or book chapters will be posted on USC Blackboard, marked as Highly Recommended / Recommended / Optional readings and covered to a large extent in the lecture materials. Some suggested readings are selected from the following books (but please note that the focus of the lecture is on current and forward-looking topics):

- R. Alur, *Principles of Cyber-Physical Systems*, MIT Press, 2015.
- T. Basten, R. Hamberg, F. Reckers, and J. Verriet, *Model-Based Design of Adaptive Embedded Systems*, Springer 2013.
- W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Elsevier, 2004.
- J. Duato, S. Yalamanchili, and L.M. Ni, *Interconnection Networks: An Engineering Approach*, Morgan Kaufmann, 2003.
- D. Kleidermacher and M. Kleidermacher, *Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development*, Elsevier, 2012.

- E.A. Lee and S.A. Seshia, *Introduction to Embedded Systems - a Cyber Physical Systems Approach*, MIT Press, 2017.
- P. Marwedel, "Embedded System Design", Kluwer 2011.
- A. Sangiovanni-Vincentelli, H. Zeng, M. Di Natale, and P. Marwedel, *Embedded Systems Development: From Functional Models to Implementations*, Springer 2013.

However, the class material will be self-contained and I can help you with suggestions to which books, chapters and articles you should read if you are interested in a particular subject as class progresses.

Prerequisite(s): The course will be self-contained and does not have formal prerequisites. However, students enrolling in the course are expected to have prior exposure to topics typically covered in an undergraduate EE program such as matrix algebra, basic graph theory, basic probability theory, basic programming, data structures and algorithms, mathematical optimization, as well as have some knowledge of control systems and communication networks.

Course Notes

Lecture slides and additional readings marked as “highly recommended”, “recommended” or “optional” will be distributed in class and also posted on Blackboard before each lecture. In addition, homework assignments, laboratory description and supplementary materials will be also posted on Blackboard before each lecture it is assigned with a clearly indicated deadline. Although the main concepts will be discussed in detail throughout the course, students are expected to read the “highly recommended / recommended” papers (and may consult the “optional” papers) in order to acquire a strong foundation on the topics covered in this class. Evaluation will be based on homework assignments, in class participation via paper presentation, midterm and a semester-long project. Students should be prepared to put in enough effort to turn in a high-quality project.

Technological Proficiency and Hardware/Software Required

Students are expected to have basic knowledge in at least one of the following high-level and general-purpose programming languages and environments: Matlab, C/C++, Python. The basic knowledge refers to programming abilities to read and modify the provided codes during the laboratory sessions and project milestones. Students are also expected to have basic knowledge of hardware design.

Description and Assessment of Assignments

The learning objectives and outcomes will be assessed through homework assignments, laboratory assignments and a semester-long project. The homework assignments assigned every two or three weeks will consist of questions and problem sets meant to test the assessment level of the knowledge discussed during the lectures. Every two or three weeks (depending on the difficulty of the homework assignment and the course workload such as upcoming project milestone or laboratory assignment difficulty level) a homework assignment is returned and a new one is posted. The questions and problems set in the homework assignments will include reading and researching the posted lecture materials and scholarly articles to assimilate the concepts discussed in class, as well as pencil-and-paper and programming exercises to achieve a comprehensive understanding. The laboratory assignments will consist of software and hardware programming problems and hardware prototyping on FPGA to acquire a concrete understanding of practical aspects of real time systems and multicore design. Also, we will design the laboratory assignments for students to get familiar with required tools for the project topics related to CPS manycore systems design. Mentoring and discussions of the homework assignment solutions as well as laboratory design exercises will be provided during office and lab hours.

The project is a major component of this course. Students can either choose their own project relevant to the course or pick one from the suggested topics. In both cases, the outcome of the project should be original research finding, well documented with regard to related work, well-supported by either theoretical proofs or experimental investigation. Students are encouraged to think big and develop out-of-the-box approaches that may lead to the development of significant solutions to problems in these areas of research. The project will count for 50% of the course grade. The project will consist of four milestones:

i) *Project definition*: Students are required to submit a 2 pages report stating the motivation for a specific project topic, discussing any prior work on solving related problem formulations or addressing challenges, outlining the problem statement, summarizing the main challenges, and formulating a tentative work plan to address the anticipated challenges. This milestone 1 will count for 5% of the course grade.

ii) *Project updates*: There are two project updates consisting of: In milestone 2, students are required to submit a 4-page report (which builds on their previous write-up) summarizing the proposed solution, discuss the various candidate strategies to solve the formulated problem and receive feedback from faculty, TA and class participants. In milestone 3, students are required to submit a 6-page report discussing the improvements developed in the project based on the feedback received in milestone 2, modifying or justifying the pursued design methodology, and describing rigorous and verified preliminary results. A special emphasis will be put on understanding the concepts discussed in class, learning the design methodologies covered in lecture material, and developing new algorithms and tools to solve the newly formulated problems in these projects. The milestones 2 and 3 will count for 10% and 15%, respectively, of the course grade.

iii) *Project evaluation*: Students are required to submit an 8-page report discussing the main results, contrasting the proposed solution with state-of-the-art solutions in terms of real-time performance and power / energy consumption, and validation and verification of the prototyped CPS design. The final milestone 4 will count for 20% of the course grade. The detailed requirements for each milestone reports and presentations will be posted on Blackboard before lecture 1.

Project presentation: Students are required to present their main project findings in an interactive session. Students will have approximately three to four weeks to work on each project milestone. Project teams of up to two or three students will be allowed, but a statement will have to be included detailing each student's contribution and assigning an agreed upon percentage contribution. The final project grade will be weighted accordingly. The course includes an overview exam which will test the assimilation of theoretical concepts discussed during the lecture hours.

Grading Breakdown

All students are required to attend all lecture and laboratory hours. Participation will not contribute towards the final grade, but missing lecture and laboratory hours will likely affect your preparation for examination. The grading scheme is as follows:

Assignment	**Points**	**% of Grade**
Homework assignments	100	10%
Laboratory assignments	150	15%
Course Project (4 milestones)	500	50%
Course examination	200	20%
Paper presentation + quizzes	50	5%
Total	1000	100

Guidelines on preparing for each individual milestone of the project, as well as the homework and laboratory assignments will contain a clear description of the points breakdown for each topic / question / problem. Failing to address a specific item in the project milestones, homework and laboratory assignment will result in losing the assigned points.

Assignment Submission Policy

No late homework or laboratory assignments will be allowed. Extensions can be granted for documented medical emergencies.

Course Schedule: A Weekly Breakdown

Week	Lecture	Assignments
August		
Mon, Aug. 26	Introduction and Course Overview. Real-time embedded systems, cyber-physical systems & Internet-of-Things: definitions, characteristics and challenges. New trends: heterogeneous, self-programming, self-organizing, self-adapting, self-optimizing, autonomous, antifragile, and cognitive CPS. Discussion of applications to be considered in class projects.	**Milestones 1**, **2**, **3** and **4** assigned; their due dates are shown below.
Wed, Aug. 28	Modeling methodologies for cyber-physical systems: Discrete, continuous and hybrid, deterministic and stochastic dynamical models, Markovian and non-Markovian dynamical models. Models of Computation in cyber and physical domains. State machines. Concurrent models of computation. Low level virtual machine (LLVM) intermediate representation. Network science characterization (suggestions for Milestone 2).	
September		
Mon, Sept. 2	**Labor Day** (university holiday)	
Wed, Sept. 4	Cyber-Physical Instruction and Task Graphs (CPI & TGs): LLVM intermediate representation of high-level programs (HLPs) and construction of weighted / hyperweighted Cyber-Physical Instruction Graphs (CPIGs) from static and dynamic compiler analysis of HLPs. Performance/energy/resource-aware automatic parallelization and construction of Cyber-Physical Task Graphs (CPTGs). Case studies: Examples of application profiling, extraction of CPIGs and CPTGs to guide the CPS manycore design.	**Laboratory assignment 1** due Sept. 20.
Mon, Sept. 9	Communication challenges for cyber-physical systems. Timing requirements, context, metrics, and variables. Heterogeneous computing for CPS: Homogeneous vs heterogeneous multicore systems, general- vs. special-purpose cores for meeting real-time requirements, bottleneck acceleration. Examples of heterogeneous multicore-based CPS. On-chip communication architectures (bus, networks-on-chip (NoC)). Structure, functionality, and design problems.	
Wed, Sept. 11	Cyber-physical systems design space exploration. Structure, functionality, and design problems. Mathematical formulation of communication infrastructure design for CPS platforms. Network science tools to tackle the computational complexity of communication infrastructure design. Greedy algorithms and sub-modularity optimization.	
Mon, Sept. 16	Project definition (part 1) – Discussion of Cyber-Physical Instruction and Task Graphs – Milestone 1 presentation, discussion, and feedback. Milestone 1 (M1) in-class presentation and discussion. Milestone 1 document distributed on Aug. 26 provides specific guidelines to prepare the in-class presentation and report: provide Application Description, Application Profiling and Characterization, Plans for Analytical Parallelization in Milestones 2 and 3.	M1 in class presentation + written report (2 pages) submitted by end of day.

Wed, Sept. 18	Project definition (part 2) – Discussion of Cyber-Physical Instruction and Task Graphs – Milestone 1 presentation, discussion, and feedback. Milestone 1 (M1) in-class presentation and discussion. Milestone 1 document distributed on Aug. 26 provides specific guidelines to prepare the in-class presentation and report: provide Application Description, Application Profiling and Characterization, Plans for Analytical Parallelization in Milestones 2 and 3.	**Homework 1** assigned and due on Oct. 2.
Mon, Sept 23	Mathematical foundations for performance analysis of real-time manycores: Discrete and continuous time Markov chains, queueing networks, examples. Throughput and latency analysis for heterogeneous manycores.	
Wed, Sept 25	Mathematical characteristics of CPS workloads, non-Markovian workload models. Throughput and latency analysis for heterogeneous manycores.	
Mon, Sept. 30	Communication infrastructure protocols: Networks-on-Chip (NoC) deterministic, adaptive and stochastic routing strategies addressing spatiotemporal computational and communication workload variability, fault-tolerance.	
October		
Wed, Oct. 2	CPS application mapping to heterogeneous computing systems: problem formulations and their mathematical properties, mixed integer linear programming algorithms and challenges, submodular optimization, application-aware mapping case studies.	**Laboratory assignment 2** due Oct. 18.
Mon, Oct. 7	Project update, demo, and discussion – part one of Milestone 2 (M2) in-class presentation and report prepared according to M2 guidelines distributed in class.	M2 in-class presentation + report (4 pages) submitted by end of the day
Wed, Oct. 9	Project update, demo, and discussion – part two of Milestone 2 (M2) in-class presentation and report prepared according to M2 guidelines distributed in class.	
Mon, Oct. 14	On-chip and off-chip interconnection design: buss, NoC and off-chip buffer sizing, virtual channel allocation. Contrasting analysis of Markovian and non-Markovian approaches to buffer sizing as a function of varying network traffic conditions. Implications for real-time scheduling.	
Wed, Oct. 16	Scheduling in CPS: Premises, problem formulation, strategies for uniprocessor task scheduling, theoretical results, limitations. Scheduling algorithms for NoC-based manycores. Open problems and outlook.	**Homework 2** assigned and due on Oct. 30.
Mon, Oct. 21	Memory trends, challenges, and opportunities. Memory hierarchy. Memory capacity gap. Optimal memory allocation with different types of memory (e.g., phase change memory (PCM), processing in memory (PIM), spin-transfer torque magnetic random-access memory (STT-MRAM)). Case study: application profiling and design of PIM architectures and validation on Hybrid Memory Cube.	
Wed, Oct. 23	Self-programming and self-optimizing manycore-based CPS: Spectral graph analysis of CPIGs and CPTGs. Processing community identification via graph curvature methods. Machine learning for self-programming and self-optimizing manycore-based CPS under uncertainty. Case studies on real codes.	

Mon, Oct. 28	Mathematical framework and algorithms for manycores reconfiguration. Network coding approaches for exascale computing. CPS code analysis, learning code representations, distributed graph-theoretic approaches to automatic parallelization.	
Wed, Oct. 30	Power and thermal management: premises & problem formulation. Workload-aware dynamic control methodologies.	
November		
Mon, Nov. 4	Project update, demo, and discussion – part one of Milestone 3 (M3) in-class presentation and report prepared according to M3 guidelines distributed in class.	M3 in-class presentation + report submitted by end of the day
Wed, Nov. 6	Project update, demo, and discussion – part one of Milestone 3 (M3) in-class presentation and report prepared according to M3 guidelines distributed in class.	
Mon, Nov. 11	Markovian vs non-Markovian optimal control strategies for power management. On-chip wireless communication, capacitive and inductive coupling chip-to-chip communication, optical interconnects for high performance computing platforms and other optimizations.	**Homework 3** assigned and due on Dec. 4.
Wed, Nov. 13	Emerging topics: Design methodologies for hardware accelerators. Bottleneck identification (code regions which due to inadequate programming models lead to violation of real-time requirements / bounds). Pre-RTL analysis and tools. Machine learning accelerator design. Design methodologies – case studies: manycores for brain activity and microbiome analysis. Large language models for formal analysis and optimization of CPS. Paper discussions.	
Mon, Nov. 18	Emerging topics: Privacy and security for ML applications. Fully homomorphic computing. Specialized hardware for machine learning. Hardware-efficient ML methods. Operating systems for embedded systems (VxWORKS, Embedded Linux, Google Android, TinyOS RTOSs). Operating systems for manycores (Barrelfish, fos, Corey, TessellationOS): incarnations, trends, challenges. Paper presentations and discussions.	
Wed, Nov. 20	Emerging topics: Tiny Machine Learning Systems. Intermittent computing. Fault tolerant methods for computation and communication in manycore-based CPS. Security challenges, analysis, and solutions. Quantitative information flow. Side channel attacks. Paper presentations and discussions.	
Mon, Nov. 25	Emerging topics: Formal analysis and verification of CPS (a special focus on probabilistic analysis and stochastic temporal logic). Complex collective concepts and swarms of CPSs. Paper presentations and discussions.	
Wed, Nov. 27	**Thanksgiving Holiday** (university holiday)	
December		
Mon, Dec. 2	Emerging topics: Trust / trustworthiness quantification of AI-based CPS. Foundations models and applications to CPS. Paper presentations and discussions.	
Wed, Dec. 4	Emerging topics: Machine learning and optimization-guided compilers for heterogeneous computing architectures in CPS. Paper presentations and discussions. Overview of the course.	

	M4 report (8 pages) electronically submitted by Dec. 10th 2024.	

Statement on Academic Conduct and Support Systems

Academic Conduct:

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, policy.usc.edu/scientific-misconduct.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
studenthealth.usc.edu/counseling
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.
National Suicide Prevention Lifeline - 1 (800) 273-8255 – 24/7 on call
suicidepreventionlifeline.org
Free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week.
Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL), press "0" after hours – 24/7 on call
studenthealth.usc.edu/sexual-assault
Free and confidential therapy services, workshops, and training for situations related to gender-based harm.
Office of Equity and Diversity (OED) - (213) 740-5086 | Title IX – (213) 821-8298
equity.usc.edu, titleix.usc.edu
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.
Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
usc-advocate.symplicity.com/care_report
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office of Equity and Diversity |Title IX for appropriate investigation, supportive measures, and response.
The Office of Disability Services and Programs - (213) 740-0776. dsp.usc.edu
Support and accommodations for students with disabilities. Services include assistance in providing readers/notetakers/interpreters, special accommodations for test taking needs, assistance with architectural barriers, assistive technology, and support for individual needs.
USC Campus Support and Intervention - (213) 821-4710. campussupport.usc.edu
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.
Diversity at USC - (213) 740-2101
diversity.usc.edu
Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call dps.usc.edu, emergency.usc.edu
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.
USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-120 – 24/7 on call dps.usc.edu
Non-emergency assistance or information.