[DOWNLOAD A USC IDENTITY GUIDELINES COMPLIANT LOGOTYPE, THEN DELETE THIS TEXT AND DRAG/DROP THE FILE HERE]

Course EE599 and Design Methodologies for Internet-of-Things Enablers: From Wearables to Data-Center-on-a-Chip Architectures

Units: 4

Spring 2018: Mon - Wed, 10:00am-11:30am (lectures)
Tue - Thu, 4:00pm-5:00pm (discussion session)
Fri, 9am-11:00am (laboratory session)

Location: TBD
Website: USC Blackboard

Instructor: Paul Bogdan
Office: Hughes Aircraft Electrical Engineering Building EEB304
3740 McClintock Ave., Los Angeles, CA 90089
USC Mail Code: 3740
Office Hours: Mon and Wed, 11:30am-12:30pm
Contact Info: Email: pbogdan@usc.edu Phone: 213-821-5720
I am replying to emails/calls as soon as I can (within hours).

Teaching Assistant: Yao Xiao
Office:
Office Hours:
Contact Info: Email: xiaoyao@usc.edu

Course Description

The laws of physics dictated not only the technological advances, but also the observed trends in embedded computing from single core to manycore systems. We are witnessing a manycore percolation in all domains from wearable, portable and medical devices, to intelligent Internet-of-Things 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 and complex bio-chemistry modeling to data analytics and deep learning algorithms are all expected to benefit from manycore systems. However, marching towards exascale computing systems is not an easy task and is poised by a number of challenges concerning the need for expressive and scalable models of computation, exact workload / application models, appropriate application to architecture performance analysis tools, efficient design and optimization algorithms of application software, the operating system and hardware. In this course, we will provide a comprehensive overview of design methodologies for manycore platforms. 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. To acquire deep knowledge in manycore design, a semester long project will focus on tackling all issues from specification, to modeling, to performance analysis and optimization.

Learning Objectives

The learning objectives of this course are: understand the benefits of developing models of computation to represent various algorithms, identify strategies for parallelization, understand the importance of accurate profiling of high level languages programs and their implications for computation, communication and memory stransactions, learn how to formulate system level optimization when considering performance, power, reliability as costs, learn how to apply theoretical concepts to concrete problems and design a manycore platform for a specific application domain, and finally learn how to communicate and present the research results and findings during in-class presentations and in milestone reports.

Required Readings and Supplementary Materials

There is no single textbook containing all subjects discussed in this class. During each lecture, a number of scholar articles or book chapters will be posted on USC Blackboard, marked as Highly Recommended / Recommended / Optional and covered to a large extent in the lecture slides. Suggested readings are selected from the following books:

- T. Basten, R. Hamberg, F. Reckers, and J. Verriet, *Model-Based Design of Adaptive Embedded Systems*, Springer 2013.
- A. Sangiovanni-Vincentelli, H. Zeng, M. Di Natale, and P. Marwedel, *Embedded Systems Development: From Functional Models to Implementations*, Springer 2013.
- P. Marwedel, "Embedded System Design", Kluwer 2011.
- J. Duato, S. Yalamanchili, and L.M. Ni, *Interconnection Networks: An Engineering Approach*, Morgan Kaufmann, 2003.
- W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Elsevier, 2004.
- D. Kleidermacher and M. Kleidermacher, *Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development*, Elsevier, 2012.

However, the class material will be self-contained and I can help you with suggestions to which books 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. Students enrolling in the course are expected to have prior exposure to matrix algebra, graph theory, basic probability theory, algorithms, mathematical optimization, as well as have some knowledge of control and communication networks. Although the main concepts will be discussed in detail throughout the course, students are expected to read the recommended papers. 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.

Course Notes

Lecture slides and recommended readings will be distributed in class and also posted on Blackboard before each class. In addition, homework assignments, laboratory description and supplementary material will be also posted on Blackboard before each homework deadline and laboratory session.

Technological Proficiency and Hardware/Software Required

Students are expected to have basic knowledge of Matlab and C programming, reading and modifying the provided codes during the laboratory sessions and 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 deifficulty of the homework assignment and the course workload such as upcoming project milestone or laboratory assignment difficulty) a homework assignment is returned and a new one is posted. The questions and problems sets in the homework assignments will include reading and researching the posted lecture materials and scholar 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 /hardware programming problems and hardware prototyping on FPGA to acquire a concrete understanding of practical aspect of multicore design. Also, we will design these laboratory assignments for students to get familiar with required tools for 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 an 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 page report stating the motivation for a specific topic, outlining the problem statement, summarizing the main challenges, discussing the related work, and formulating a tentative work plan to address the anticipated challenges.

ii) *Project update*: Students are required to submit a 4-page report (which builds on their previous write-up) summarizing the proposed solution and some preliminary results.

iii) *Project evaluation*: Students are required to submit an 8-page report discussing the main results and contrasting the proposed solution with state-of-the-art solutions.

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 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 result in lower changes to obtain a high grade. The grading is as follows:

Assignment	Points	% of Grade
Homework assignments (up to 7)	100	10%
Laboratory assignments (up to 5)	150	15%
Course Project (4 milestones)	500	50%
Course examination	200	20%
Paper presentation + quizzes	50	5%
Total	1000	100

Grading Scale (Example)

Course final grades will be determined using the following scale

A 95-100
A- 90-94
B+ 87-89
B 83-86
B- 80-82
C+ 77-79
C 73-76
C- 70-72
D+ 67-69
D 63-66
D- 60-62
F 59 and below

Assignment Submission Policy

No late homework or laboratory assginments will be allowed.

Course Schedule: A Weekly Breakdown

	Topics/Daily Activities	Readings and Homework
Week 1	Embedded systems & Internet-of-Things: definitions, characteristics and challenges.	**Homework 1 assigned** **Laboratory 1**
Week 2	Models of computation (finite state machines, Kahn processes, statecharts). Low level virtual machine intermediate representation and application profiling tools.	**Project Milestone 1**
Week 3	Mathematical models for application task graphs and workloads (synthetic and realistic benchmarks). Discussion of applications considered in class project.	**Homework 2 assigned** **Laboratory 2**
Week 4	Memory hierarchy. Memory capacity gap. Examples (DRAM, phase change memory) and challenges.	
Week 5	On-chip communication architecture: alternatives (bus, networks-on-chip) and problems	**Homework 3 assigned** **Laboratory 3**
Week 6	Real-time models and performance analysis for heterogeneous manycores	**Project Milestone 2**
Week 7	Networks-on-Chip: topology synthesis, buffer sizing, virtual channels allocation	**Homework 4 assigned** **Laboratory 4**
Week 8	Networks-on-Chip: routing protocols, application-aware mapping and scheduling	
Week 9	Mathematical framework and algorithms for manycores reconfiguration. Network coding approaches for exascale computing.	
Week 10	Design methodologies for hardware accelerators: Pre-RTL analysis and tools	**Homework 5 assigned** **Laboratory 5**
Week 11	Power and thermal management. Workload-aware dynamic control methodologies	**Project Milestone 3**
Week 12	Fault tolerant methods for computation and communication in manycores. Security	
Week 13	Operating systems for embedded systems (VxWORKS, Embedded Linux, Google Android, TinyOS RTOSs) and manycores (Barrelfish, fos, Corey, TessellationOS): incarnations, trends, challenges	
Week 14	On-chip wireless communication, capacitive and inductive coupling chip-to-chip communication, optical interconnects for high performance computing platforms	
Week 15	Emerging topics. Paper presentations and discussions. Overview of the course	
FINAL	Final Project Evaluation – In class presentations of semester long projects and submission of project reports	**Project Milestone 4**

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, http://policy.usc.edu/scientific-misconduct.

Support Systems:

Student Counseling Services (SCS) – (213) 740-7711 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention. engemannshc.usc.edu/counseling

National Suicide Prevention Lifeline – 1 (800) 273-8255
Provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week. www.suicidepreventionlifeline.org

Relationship and Sexual Violence Prevention Services (RSVP) – (213) 740-4900 – 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender-based harm. engemannshc.usc.edu/rsvp

Sexual Assault Resource Center
For more information about how to get help or help a survivor, rights, reporting options, and additional resources, visit the website: sarc.usc.edu

Office of Equity and Diversity (OED)/Title IX Compliance – (213) 740-5086
Works with faculty, staff, visitors, applicants, and students around issues of protected class. equity.usc.edu

Bias Assessment Response and Support
Incidents of bias, hate crimes and microaggressions need to be reported allowing for appropriate investigation and response. studentaffairs.usc.edu/bias-assessment-response-support

The Office of Disability Services and Programs
Provides certification for students with disabilities and helps arrange relevant accommodations. dsp.usc.edu

Student Support and Advocacy – (213) 821-4710
Assists students and families in resolving complex issues adversely affecting their success as a student EX: personal, financial, and academic. studentaffairs.usc.edu/ssa

Diversity at USC
Information on events, programs and training, the Diversity Task Force (including representatives for each school), chronology, participation, and various resources for students. diversity.usc.edu

USC Emergency Information
Provides safety and other updates, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible. emergency.usc.edu

USC Department of Public Safety – UPC: (213) 740-4321 – HSC: (323) 442-1000 – 24-hour emergency or to report a crime.
Provides overall safety to USC community. dps.usc.edu