SYLLABUS

COSC-520, Computer and System Architecture, spring 2012 (3 units)
Computer Science Department
Georgetown University

Lecture 9:30-10:45 am TR White-Gravenor 208

Prof. Richard K. Squier
St. Mary's Hall, 339
phone: 202-687-6027
squier@cs.georgetown.edu
office hours: Tue/Thu 11:00-12:00

COURSE DESCRIPTION:

This course is about the evolution of the design of computer systems and the factors shaping their hardware/software system architectures. We begin with a survey of information theory and physical information, and trends in device and systems technology. We review basic computer system hardware architecture, implementation, organization, functionality, general features of instruction sets, parallel architectures, and general parallelism in both software and hardware. We look at some specific system architectures at various scales from embedded computer systems to warehouse computing. Topics include Moore's Law; Amdahl's law; Flynn's taxonomy; silicon device properties; system power, performance, dependability, and cost; parallelism in processor architecture; memory hierarchies (cache and virtual memory); processor micro-architecture and pipelining; protection and sharing; I/O and interrupts and exceptions; in-order and out-of-order superscalar architectures; VLIW machines; vector supercomputers; multi-cores; clusters; multi-threaded architectures; symmetric multi-processors; parallel computational models and programming; GPU and special-purpose processors; reconfigurable logic and architectures; and warehouse-scale hardware, file systems, and computing. A systems design project implements processor ILP mechanisms and memory hierarchy. Grading: participation, 15%; midterm, 20%; project 35%; final, 30%. Extra credit can be given for exceptional project development or other activities with instructor's permission.

Prerequisites: Expected background includes an understanding of computer systems from basic hardware organization through operating systems at the undergraduate level. Undergraduates may enroll with instructor's permission.

Required text: Hennessy and Patterson. *Computer Architecture, A Quantitative Approach, 5th ed*. ISBN 978-0-12-383872-8. Morgan-Kaufman, 2012.

Recommended background text: Patterson and Hennessy. *Computer Organization and Design, The Hardware/Software Interface, 4th ed. Revised*. ISBN 978-0-12-374750-1. Morgan-Kaufmann, 2011.

READINGS

Aftab-Information Theory and the Digital Age-2001

Frank-PhysLimitsComp-2002

Krone-FundatmentalPhysicalLimitInfoTransAndProcess-2010

Fuller-ComputingPerformanceGameOverOrNextLevel-2011
Larus-MooresDividend
Nagy-SuperexponentialLongTermTrendsInInfoTech-2010
Yao-ExtendingAmdahlLawInMulticoreEra-2009
Ranganathan-RecipeForEfficiencyPrinciplesOfPowerawareComputing-2010
Russell-CRAY_1_ComputerSystems-1978
CrayXT6Brochure-2010
AMD-AMDmulticoreProcessors-2006
Schlansker-EPIC_ArchitectureFor_ILP_ParallelProcessors-2000
Shalf-NewLandscapeOfParallelCompArch-2007
Herbodt-ComputingModelsForFPGA-2008
Wimmer-ProgrammingModelsForParallelComputing-2010
McCool-ScalableProgrammingModelsForMassivelyMulticore-2008
Buchty-SurveyHeteroMulticoreAndAccelerators-2009
Ghemawat-GoogleFileSystem-2003
Barroso-DatacenterAsComputerIntrotoWarehouseScaleArch-2009
Foster-CloudGridComputingCompared-2009
Takabi-SecurityAndPrivacyChallengesInCloud-2010
IBM-UnderstandingDRAM_Performance-1996
McNerney-RoleOfDesignComplexityInTechImprovement-2009
Guz-ManyCoreVsManyThread-2009
Doursat-ProgrammableArchSelfOrganized-2008
Barham-XenAndArtOfVirtualization-2003
Killalea-MeetTheVirts-2008
Mudge-PowerFirstClassArchDesignConstraint-2001
Brown-TowardEnergyEfficientComputing-2010
Woo-ExtendingAmdhalLawForEnergyEfficientManycore-2008.pdf
Ambs-OpticalComputingA60yearAdventure-2010
Dressler-SurveyOnBioinspiredNetworking-2010
Pershin-PracticalProgrammableAnalogCircuitsWithMemsistors-2009
Baumann-MultikernelNewOSarchForScalableMulticoreSystems-2009
Ranger-EvaluatingMapReduceForMulticoreAndMultiprocessor-2007
Castro-FundamentalsOfNaturalComputingOverview-2007
Lindley-BrainsAndBytes-2010

ACADEMIC INTEGRITY POLICY:

In assigning grades, one of my jobs as instructor is to ascertain your growth in understanding the intellectual content of the course during our studies together. Course assignments and projects are intended to facilitate that growth. However, at times, one's thinking can get lead astray by side-issues that may seriously hamper your efforts to understand. It is very important that you do not dwell fruitlessly on some point that has you stuck. You should seek help as soon as practical, and your classmates can be an efficient resource. For that reason, I encourage you to freely exchange information, and this Academic Integrity Policy is designed to allow for, and encourage that kind of cooperation. The default policy for the Computer Science Department is amended as follows. You are free to discuss problems and solutions of any assignment or project or exam with your classmates or others. You need not cite these conversations nor indicate which parts of your submitted material was garnered from such conversations. You are free to collect information from any source, electronic or otherwise, and you need not indicate the original source nor that the material did not originate with you. In addition, in this context, I consider it a fault to withhold useful information from others; although, this policy makes no stricture against it. The ability to work cooperatively together is a learned skill that will be important later in life, and sharing information is critical.

GRADING:

My grading system does not depend on evaluating your progress based on material of unknown origin. Homework is graded, but used solely to provide feedback, and not in determining grades (However, see class participation.) I do use your submitted material as a guide in developing examinations, and may ask that all your work be returned to me temporarily; so, keep ALL your work together as a portfolio. If you feel you are not being evaluated thoroughly enough, it is incumbent on you to bring this to my attention while there is still time to address your concerns before grades are submitted. You are welcome to discuss these issues with me at any time.

HOMEWORK MARKUP SYSTEM:

Homework grading should be thought of as a discussion, rather than a score. We mark up your work and return it to you. You can, and should if you feel it worth it, return your work with comments. I might then follow up in class or with additional comments, or you might follow up in class.

A check mark means:
"I mostly agree with what you said", "I cannot find anything worth quibbling about"

A “-" means :
"I think you are about 1/2 right", "There is something missing here, but not entirely wrong"

An “X” means:
"You did not answer the question", "The answer was too skimpy", "You basically missed the point"

A “?” means:
"I do not understand what you said", "Are you sure this makes sense?"

A “+" means:
"I like what I see", "You went beyond the call of duty", "Nice job".