EE/CpE-517A Computer and Digital Systems Architecture

Catalog Description:

EE/CpE-517A Computer and Digital Systems Architecture
This course covers the design and architecture of computer and digital systems in the system design region above the transistor/logic gate level and below the device driver level/system monitor level. The goal is to enable students to understand design considerations and tradeoffs in this middle level that EE/CpE/CS students often take for granted. The systems considered in the course will go beyond the computer chips or CPUs discussed in a typical computer architecture course, but will include complex logic devices such as application specific integrated circuits (ASICs), the core-designs for field programmable gate arrays (FPGAs), system-on-a-chip (SoC) designs, ARM, and other application-specific architectures. Printed circuit board-level architectural considerations for multiple complex digital circuits will also be discusseD

Text Book:

Steve Furber, "ARM System-on-chip Architecture," 2nd ed., Addison-Wesley, 2000, ISBN 0-201-67519-6
optional: Patterson and Hennessey, Computer Organization and Design, 4th Edition, Morgan Kaufmann, 2009, ISBN: 978-0-12-374493-7.

Instructor:

Bruce McNair, Distinguished Service Professor of ECE.

Goals:

The goal of this course is to give students an understanding of the issues involved in designing complex digital systems, including computer systems and special purpose processing devices.

Course objectives:

This course will enable students to understand how the computing and digital system infrastructure used in a wide variety of systems is designed. Students will learn to architect and design complex digital systems. They will be able to understand the tradeoffs between hardware and software design options in embedded systems and will be able to put their electrical and computer engineering course work in the context of practical designs of real systems.

Course Outcomes:

Course Outcome 1: The student will be able to understand the architectural tradeoffs involved in designing internal data paths for a computer or digital system.

Course Outcome 2: The student will be able to understand the hardware requirements imposed by instruction set choices for a digital computer.

Course Outcome 3: The student will be able to understand physical limitations that determine the operating speed of computer and digital systems and the implication of these limitations on memory organization.

Course Outcome 4: The student will be able to understand the constraints placed on a computer or digital system by the manner in which it is programmed, controlled or interfaces with by external systems.

Course Outcome 5: The students will be able to understand how modern digital system combine processing elements with general **purpose** fixed or programmable logic elements

Prerequesites by Topic:

- Computer organization
- Electronic circuit fundamentals
- Familiarity with programming in a high-level language (e.g., C, Pascal)
- Basic understanding of switching theory and logical design

Grading Policy:

Participation in class discussions 10%
Homework 50%
Written project 35%
Project presentation: 5%

All assignments provide opportunities for extra credit work. Work that goes significantly beyond what is asked will be graded accordingly.

This is a graduate course. As such, the student privileges accorded students by the Stevens Honor System do not apply. However, student responsibilities for proper citation of references, academic integrity, and professional conduct in completing course work are assumed. Any deviation from the spirit of the Stevens Honor System and/or the graduate academic integrity policies is unacceptable.

Course Components:

- Engineering - 100%

Course Web Site:

http://koala.stevens-tech.edu/~bmcnair/CpE517-XXX where XXX is the current semester, e.g., F12

Schedule of Topics

Week 0: Introduction, logistics, course structure

Week 1: Digital electronics, digital logic, computer organization and introduction to processor design

Week 2: General computer and digital system design considerations: e.g., arithmetic/logic functions, signal processing, real-time control, etc.

Week 3: ARM architecture

Week 4: ARM organization and implementation

Week 5: ARM instruction set, instruction set tradeoffs

Week 6: ARM support for high-level languages

Week 7: Architectural support for system development

Week 8: ARM processor cores

Week 9: Hierarchical organization of system memory

Week 10: Architectural support for operating systems

Week 11: FPGA cores and ARM CPU cores

Week 12: Embedded ARM applications

Week 13/14: Project presentations

Last revised: August 27, 2012