ELEC4310 Embedded System Design

Course Description

In this course, students will learn the important concepts and modern design practices of embedded computing systems. They will see how a complex embedded system can be systematically developed as a union of software and hardware. The course will cover several fundamental topics, such as design targets, hardware/software co-design methodology, common design techniques, processors, architectures, and physical implementations. It will also cover several advanced topics, such as behavioral modeling, low-power techniques, and systems-on-chip. *Prerequisite(s):* [COMP 2611 AND (ELEC 2200 OR ELEC 3310)] OR ELEC 2300 OR ELEC 2350 OR ISDN 4000F

List of Topics

Lecture Topics

Week 1	**Introduction to Embedded Systems (I)** ES History, Social Impacts, Applications, Definition, ES Industry, Characteristics, Design Metrics, Performance (Peak, Average, Worst-case, Response Time, Throughput), Energy Efficiency, Size in ASIC/FPGA/GPP/ASIP, NRE, Unit Cost
Week 2	**Introduction to Embedded Systems (II)** Design Categories, Processor Architectures and Comparison (GPP, ASIP, SPP), Physical Implementation (Multichip, Single Chip, Full-custom, Gate Array, Standard Cell, FPGA, Chip Cost), Design Automation, Example (Google Phone)
Week 3	**SoC and ES Development Platform** SoC Definition, SoCs in iPhone 1 to 4Gen, SoC Subsystems, SoC Example (TI OMAP3530), ES Hardware Development Platform, ES Software Development Environment (IDE), Debugging and Testing (ISS, ICE, JTAG)
Week 4	**HW/SW Codesign Methodologies (I)** Methodology Overview, TTM, Man-month Dilemma, Application Specifications, Behavioral Modeling, Computational Models, Modeling Languages (Textual, Graphical, Software-oriented, Hardware-oriented), Models vs. Languages
Week 5	**HW/SW Codesign Methodologies (II)** Sequential Program Model, Formal FSM, FSMD, HCFSM, PSM, Concurrent Process Model, Dataflow Model, SDF
Week 6	**HW/SW Codesign Methodologies (III)** HW/SW Partitioning and Mapping, Formal Verification (Rule of 10X, Correctness, Completeness), Testing (JTAG, BIST), Performance and Power Evaluations, Simulation Levels and Comparisons
Week 7	**Embedded GPP and ASIP (I)** Processor and Memory Architectures (Harvard, Princeton), Examples (ARM Cortex, Intel Atom), RISC, CISC, VLIW, ES and Desktop GPP Comparisons
Week 8	**Embedded GPP and ASIP (II)**

	ASIP, Microcontroller (8051, 68HC11), DSP (StarCore, TriMedia), Customizable ASIP (Xtensa), IP (classification, licensing models, protection), GPP/ASIP Selection, GPP/ASIP Market
Week 9	**Embedded GPP and ASIP (III)** ES and Desktop Programming Comparisons, Concurrent Processes, Inter-process Communications (Message Passing, Shared Memory), Critical Section, Mutex, Deadlock, Process Synchronization, Condition Variable, Monitor
Week 10	**Embedded GPP and ASIP (IV)** Embedded OS, Multitasking, Execution Time, Deadline, Process Period, Thread, Process Scheduling, Priority Queue (Rate Monotonic, Deadline Monotonic), POSIX, RTOS and Examples (WinCE, QNX), Android
Week 11	**Single-Purpose Processor (I)** SPP and GPP Similarities, SPP Applications, Behavior Model, SPM to FSMD Conversion, Data Path Design, Control Path Synthesis, RTL, Standard Cell-based Backend Design, Computational Model Optimizations, Datapath Optimizations
Week 12	**Single-Purpose Processor (II)** Power Consumption Models and Trends, Dynamic Power, Activity Factor, Short-circuit Power, Leakage Power, Low-power Technique Classification and Effectiveness, Technology Mapping, State Encoding, Path Balancing, Retiming, Clock Gating, DFS, DVS, DVFS
Week 13	**Single-Purpose Processor (III)** Voltage-based Leakage Reduction Techniques, Power Gating (Coarse-grain, Fine-grain), Standby, Gate Sizing, Multi-threshold Logic, VTCMOS, Isolation and Voltage Shifter, Voltage Drop, Rush Current, ES Power Management
Week 14	**Industrial Talks** Talks about ES Design Cases by Engineers and Managers

Lab Topics

1. ES Development Platform
2. OMAP Development Platform
3. Qt Applications
4. Qt Components
5. OpenCV
6. Smart Camera
7. Google Android

<u>Statement of Objectives/Outcomes:</u>

On successful completion of this course, students will be able to:

CO1 - understand the definition and importance of embedded systems, and its relationships with other systems

CO2 - understand typical embedded system design methodologies, especially hardware/software codesign flow

CO3 - partition embedded applications into hardware and software, and develop architectures for common embedded systems

CO4 - systematically design individual hardware components, such as application-specific processors, for embedded systems, and reuse hardware intellectual properties

CO5 - systematically develop software components for embedded systems, and reuse software intellectual properties

CO6 - use typical embedded system platforms, especially ARM-based system, to design, implement, and test embedded systems

Textbook(s):

No required textbook.

References

1. Frank Vahid and Tony Givargis, *Embedded System Design: A Unified Hardware/ Software Introduction*, John Wiley & Sons, 2002
2. Wayne Wolf, *Computers as Components, Principles of Embedded Computing System Design*, Morgan Kaufmann Publishers, 2001

Relationship of Course to Program Outcomes:

Please refer to the Report Section 4.3.2 (iii).

Grading Scheme:

Homework	10%
Labs	30%
Midterm Examination	30%
Final Examination	30%