

Hardware-Software Interface

EE 4511, EE 5511

EE 4561 (Lab)

University of Colorado Denver
College of Engineering and Applied Science
Department of Electrical Engineering

Term: Fall 2009

Meeting: M,W 5:30-6:45 (W 7-9:45 lab)

Course location: NC 2408, NC 2609 (Lab)

Office Hrs: T/Th 10am-12noon, W 2pm-4pm

Professor Dan Connors

Email: Dan.Connors@ucdenver.edu

Office: NC 2204A

Website: Web material will be posted at BlackBoard (EE4511): <http://blackboard.cuonline.edu>

NOTE: All email communication by students must use ucdenver.edu as the email domain, emails from gmail, hotmail, yahoo, etc are NOT considered valid methods of communication.

Course Design

Catalog Description:

Computer engineering methods in hardware and software design applied to problems of modern computer systems. Interface and real-time programming techniques are considered. Graduate level (cross-listed EE 5511) requires additional projects and homework.

Instructor Description:

The course covers the design principles of modern microprocessor-based computer systems, with a specific focus on using System-on-chip (SoC) development environments (compilers, soft IP cores, FPGA prototyping). Lectures cover both hardware and software aspects of microprocessor system design such as interfacing to input and output devices, user interface design, real-time systems, and table-driven software. The homework emphasis is on software development, especially interfacing with hardware, in assembly language.

Prerequisites:

EE 1510 Logic Design, EE 1520 Embedded Systems I, EE 2520 Embedded Systems II, and EE 3651 Digital Hardware Design.

Topic Prerequisites:

Knowledge of a high level language; understanding of computer data structures; knowledge of stack mechanisms and procedure calls, understanding of assembly language programming (op codes, operands, etc.); boolean logic design techniques; realization of boolean functions using AND, OR, XOR, NAND, NOR gates and appropriate minimization techniques; logic building blocks (flop-flops, multiplexors, decoders, shift registers), design and implementation of finite state machines; familiarity with modern digital CAD tools to implement and simulate digital designs, knowledge of Verilog (or VHDL) design language, knowledge of Linux commands and file system, and experience with compiler (i.e. gcc) and debugging environments.

Course Objectives: Learning Hardware/Software Co-Design Principles and Methodology

Hardware Design

- Understanding the architecture of a typical development board and functionality of on-board devices
- Understanding of the Altera chip architecture and the functional of its components
- Understanding of the Altera internal microprocessor Bus Architecture
- Defining a system architecture using the Altera FPGA chip and the on-board modules
- Designing and implementing a custom hardware and its interface with the bus architecture

Software Design and Using State-of-the-Art Design Environments

- Understanding the details of system software components (boot loading, initialization, system stack, system calls) and interfacing hardware with high-level and low-level techniques
- Understanding the capabilities of Quartus II Design Environment for design and verification.

System-on-Chip Design

SoC designs result from the increased density of functionality that can be placed on a single chip. SoC designs are characterized by functional complexity which can not be effectively accommodated by traditional hardware-only design methods. The embedded computer system architectures on these chips include concurrent software executing on one or more processors, operating system schedulers, and hardware models, including application specific functionality, buses and networks. Through lectures, readings, presentations, discussions, and projects, this course presents the fundamental models and design steps that enable the design of SoC from the software through the logic levels, with the goal of integrating multiple components and other systems.

At the end of this course, students will be able to:

- Understand the methodology to specify, design and implement a microprocessor based SoC.
- Employ computer aided design tools to design and implement a microprocessor based SoC.
- Understand the concepts behind hardware/software co-design and the advantages in using hardware/software co-verification tools.
- Calculate the specifications and bandwidth requirements of a given embedded system.

Sample Altera-based Design Description
Introduction to the Altera Nios® II Soft Processor
Introduction to the SOPC Builder
Using the SDRAM Memory on Altera's DE2 Board
Using C with Altera DE2 Board
Debugging of Application Programs on Altera's DE2 Boards

Laboratory Objectives:

To develop skills in designing the hardware of a SoC including: programming, debugging, interfacing hardware component.

Sample Lab Description
Lab 1: A Simple Computer System
Lab 2: Program-Controlled Input/Output
Lab 3: Subroutines and Stacks
Lab 4: Polling and Interrupts
Lab 5: Bus Communication
Lab 6: Bus Control and Arbitration
Lab 7: DMA Data Transfer

Requirements

Assessment Design/Grades:

- (15 %) Exam I
- (15 %) Exam II
- (25 %) Final
- (20 %) Homework Assignments
- (25 %) Lab Exercises

Grades are as follows:

- A – “Superior/Excellent”, 90 – 100%
- B – “Good/Better than Average”, 80 – 89%
- C – “Competent/Average”, 70 – 79%
- D – “Minimum Passing”, 60 – 69%
- F – “Failing”

Textbook:

No required textbook for the course. However, material will be posted from several sources. Students are responsible for printing material for their own use.

On-line material for the Altera DE2 development platform and architecture:

DE2 Basic Computer System User Manual

<http://www.altera.com/education/univ/support/examples/unv-example-systems.html>

There are 4 free on-line reference handbooks:

- Nios II Processor Reference Handbook (Nios PRH)
- Nios II Software Developer’s Handbook (Nios SDH)
- Embedded Peripherals (EP)
- Embedded Design Handbook (EDH)

All are available at: <http://www.altera.com/literature/lit-nio2.jsp>

The material is available on a per-chapter basis, allowing students to download and print specific concepts. The handbook acronyms (Nios PRH, Nios SDH, EP, and EDH) will be used to assign reading.

Assignments and Examinations:

Examinations are intended to measure your individual mastery of the material. Exams concentrate on your understanding of the important concepts, rather than your ability to memorize details. All major examinations will be held in class with exact dates determined in class. The exams will generally test your knowledge of assignment material, so you are responsible for mastering all lab, homework, and programming material submitted with other partners, as if you did all the work by yourself. All exams will be open book and open notes (unless otherwise stated). The nature of the course material is such that the final exam must be cumulative.

Lab Exercises: Laboratory exercises are meant to develop design and implementation skills in both hardware and software. These exercises will be due every 1-2 weeks.

Course Policies: Policies regarding class attendance, turning in late work, missing homework, tests or exams, make-ups, requesting extensions, reporting illnesses, cheating and plagiarism, changes to the syllabus. Academic policies will be consistent with the University's policies at the College of Engineering and Applied Science's website: <http://www.cudenver.edu/Academics>

Class environment:

No phone calls, no text messaging, and no operation of computers for web reading, chatting, emailing, or gaming will be tolerated. Using computers to visit websites distracts other students. Use of computers to read lecture notes and power point slides is appropriate. No loud talking between students is allowed. The goal is to create an excellent environment for learning and teaching.

Extensions/make-ups:

In general, late work will not be accepted. Turn in all work by the established deadline. In case you have difficulties finishing an assignment contact the instructor before the deadline. Late work can be accepted only under circumstances beyond student's control and after arrangement with the Instructor, prior to the deadline. Note: work turned-in on time is eligible for partial credit. It will always be better to turn work in by the deadline, as trying to "perfect" it and turn it in late will give you no points at all. You have to follow the submission and media policies and guidelines published on the web. Plagiarism is the passing of someone else's work as one's own, without giving the original author due credit. Scholastic dishonesty will be treated very strictly as per University of Colorado Denver rules.

Lecture:

Lecture material (slides,notes) will be made available on the web prior to class. Lecture will also consist of chalk drawings, overhead drawings, and content not explicitly present in slides and notes.

Laboratory Resources: NC2408 and NC2609 each contain 15 working Core2-Duo stations. Each station has been loaded with Quartus® II development software version 9.0. Students are invited to download the web edition software to their home computer or laptop:

<http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html>

Unfortunately, no outside university support will be offered to install the software on the student's home system. The software has particular platform requirements. There are 30 Altera DE2 development boards available for the course. Students are expected to responsibly treat the development boards and station.

Students called for military duty

If you are a student in the military with the potential of being called to military service and /or training during the course of the semester, you are encouraged to contact your school/college.

Tentative Course Schedule

Week	Lecture
1	Introduction to embedded processor design and architecture.
2	Nios II Architecture: instruction set, assembly language, assembler. Program development in C.
3	Polling, interrupts (interrupt service routines), I/O configuration, serial and parallel ports.
4	Bus and interconnect architectures, IP cores, FPGA synthesis.
5	Exam (checkpoint) and memory hierarchy.
6	Software development.
7	Timers, design timing-constraints and requirements.
8	Monitors, in-circuit emulators, debuggers, monitors, software engineering, debugging using software.
9	Design optimization trade-off analysis.
10	Exam (checkpoint)
11	Hardware-software specialization
12	Network-interconnect
13	Emerging technologies.
14	Project development.
15	Project development.
	FINAL EXAM PERIOD

Altera DE2 Development Information

The complete DE2 package includes An 8-inch by 6-inch DE2 board with a Cyclone II EP2C35 (672-pin package) FPGA,,Quartus® II Web Edition software, Nios® II embedded processor

Figure 1. DE2 Development and Education Board



DE2 Board Information	
Feature	Description
FPGA	Cyclone II EP2C35F672C6 with EPCS16 16-Mbit configuration
I/O Devices	Built-in USB-Blaster™ cable for FPGA configuration 10/100 Ethernet RS232 Video Out (VGA 10-bit DAC), Video In (NTSC/PAL/Multi-format) USB 2.0 (type A and type B) PS/2 mouse or keyboard port Line In/Out, Microphone In (24-bit Audio CODEC) Expansion headers (76 signal pins) Infrared port
Memory	8-MBytes SDRAM, 512K SRAM, 4-MBytes Flash SD memory card slot
Displays	16 x 2 LCD display and Eight 7-segment displays
Switches and LEDs	18 toggle switches 18 red LEDs 9 green LEDs Four debounced pushbutton switches
Clocks	50 MHz crystal for FPGA clock input 27 MHz crystal for video applications External SMA clock input