Course Outline

COMP4601
Design Project B
Teaching Period 1 2011

Contents

- Course Staff
- Course Details
- Course Summary
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Staff

Contact Details

Staff Name	Role	Email	Phone
Oliver Diessel	Course Convenor	odiessel at cse.unsw.edu.au	57384

Consultations

Please email me to arrange a suitable meeting time.

Course Details

Course Code:	COMP4601
Course Title:	Design Project B
Units of Credit:	6
Course WebSite:	http://www.cse.unsw.edu.au/~cs4601/
Handbook Entry:	http://www.handbook.unsw.edu.au/undergraduate/courses/2011/COMP4601.html

Course Summary

COMP4601 is a team-based project development course involving the design and implementation of embedded and high performance or low power systems using FPGAs.

The 2011 project involves the development of an image processing system for a vision prosthesis. In outline, the project entails:

- converting a PAL video camera stream into RGB for display on a VGA monitor,
- processing the video stream to produce a rendering of the signals to be sent to an implanted device, and
- providing methods for controlling various aspects of the processing.

Project teams are expected to document their design, write a report describing their development and present their solution.

See the 2011 COMP4601 Project Description for details.

Lectures will be provided in part to guide the design of the project solution and to provide the context within which configurable logic design is undertaken. Lectures will cover:

- motivations for hardware-based computer vision,
- video formats and video conversion,
- overview of FPGA-based systems and technology,
- FPGA-based computing techniques,

- implementing configurable systems, and
- case studies of configurable image processing systems.

Course Aims

This course aims to introduce students to the design and implementation of digital systems using configurable hardware. Students will work in teams to develop an FPGA-based solution to a compute intensive or low-power problem. The course aims to provide an understanding of the usefulness and limitations of custom logic as an alternative to processor-based computing.

Student Learning Outcomes

By the end of this course, students should be able to:

- design and implement digital systems using configurable logic,
- be able to distinguish when processor-based and logic-based solutions to computing problems are called for,
- explain and put into practice techniques for accelerating computation in hardware,
- independently study advanced techniques in digital design and implementation, and
- document and present their developments in a professional manner.

This course contributes to the development of the following graduate attributes:

Graduate Attribute	Where Acquired
an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context	Design & implementation tasks
the capacity for enterprise, initiative and creativity	Design & implementation tasks
the skills required for collaborative and multidisciplinary work	Team activities
the skills of effective communication	Presentation & documentation tasks

Assumed Knowledge

The formal pre-requisites for this course are COMP3222 and COMP3601.

It is assumed that, prior to taking this course, students:

- know how to program an FPGA,
- know how to construct combinational and sequential digital circuits, and
- understand the concepts of sequential programming i.e. be familiar with the Von Neumann model of sequential computation and know an imperative programming language such as C.

Teaching Rationale

Hardware design and systems prototyping using field-programmable gate arrays is increasingly important and supports a booming embedded systems industry. In order to develop essential skills and to be industrially relevant, Computer Engineering students need to gain experience designing and implementing systems and components using programmable logic devices. The background necessary to understand the design problem and programmable logic technology will be outlined in classes. Students will work in teams to study and develop solutions using current methodologies to a computationally demanding and relevant problem.

Teaching Strategies

Formally, the course comprises two + one lecture hours and two lab hours per week. However, since this is a project course, there will be an emphasis on working informally with your team members to design solutions to the given problem.

Topics on configurable hardware and computing, and topics relevant to the project design, will be introduced and discussed during lectures. Attendance, particularly during the Tuesday lecture time, is expected.

The lecturer will meet with each group during their usual lab session in Weeks 5 and 10 to review progress on the project design and implementation. Attendance at these meetings by all team members is compulsory.

Your team will be scheduled to have use of the Lyre (ASyst) lab on the Ground Floor of K17 for one of the two two-hour sessions from 9-11 am on Tuesdays or 10-12 pm on Wednesdays. You are welcome to use the lab at other times when it is not being used for another scheduled class and room is available. However, you are guaranteed space during your scheduled lab session. Lecturing and demonstrating staff will be on hand during your scheduled lab session for consultation and assistance. Attendance is voluntary.

Teams will demonstrate their system and present their project work to the class during Week 13.

Given the complexity of teaching a group of students with diverse backgrounds, the lecturer may revise or take detours from the plans stated in this Course Outline in order to better achieve the course objectives.

Assessment

Assessment in this course will be based on the following components:

- Project Design and Implementation (45%), a team-based score,
- Documentation (30%), a team-based score, comprising 10% for the two review meeting reports and 20% for a technical report describing the team's project submission,
- Week 13 Presentation (10%), a team-based score, and
- Peer Assessment (15%), an individual score.

See the 2011 COMP4601 Project Description for details.

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Schedule

An overview of topics that will be presented in lectures is provided during the Introductory lecture in Week 1.

See the 2011 COMP4601 Project Description for a timeline of tasks to be completed for the project.

Resources for Students

There is no prescribed text for this course. However, students will need to refer to product data sheets, standards, system documentation, reference books and technical articles to gain the background needed to design and implement the project system. Some of these references will be identified during the course.

Course Evaluation and Development

Student feedback on this course will be obtained via electronic survey (CATEI) at the end of session and will be used to make improvements to the course. Students are encouraged to provide informal feedback during the session and to let the lecturer in charge know of any problems as soon as they arise. Every reasonable effort to address concerns will be made.

Several changes have been introduced into the 2011 offering of the course in response to feedback from the second offering of the course in 2010. These changes include: providing more assistance with simulating VHDL designs; providing a memory controller wrapper and test bench to facilitate interfacing with DDR memory; providing more explanation of the teaching philosophy being followed in the course and more closely aligning the teaching materials with the desired project outcomes.