Course Outline

COMP3211 / COMP9211
Computer Architecture

Teaching Period 1 2012

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
Sri Parameswaran	Lecturer	sridevan at cse.unsw.edu.au	57223

Consultations

Staff Name	Day/Time	Location
Oliver Diessel	Tuesday 1500-1600	Please email me
Oliver Diessel	Wednesday 1500-1600	Please email me

Course Details

Course Code: COMP3211 / COMP9211
Course Title: Computer Architecture
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs3211/
Handbook Entry: http://www.handbook.unsw.edu.au/undergraduate/courses/2012/COMP3211.html

Course Summary

In this course students study the architecture & organisation of modern processors, and influences upon these, with emphasis on pipelined Reduced Instruction Set Computer (RISC) machines. Students will gain an understanding of the design of the memory subsystem, I/O, and the system level interconnect. Students become proficient in the use of tools and techniques for the description, simulation, and verification of architectural designs. Students complete a sequence of assignments leading to the design, implemention, validatation and assessment of a RISC system. It is assumed students are familiar with combinational and sequential logic design principles and have experience in the use of CAD tools to describe, simulate and implement digital systems.

Course Aims

The objectives of this course are to:

- develop a deeper understanding of computer systems design with emphasis on pipelined RISC machines;
- become familiar with basic design principles and techniques for performance improvement;
- know how to apply simulation tools and appreciate the importance of simulation as the primary means of validating designs; and
- practice professional skills in design and analysis, project management, and presentation.

The course and its aims relate to other fields of study within the discipline in the following ways:

- Computer Architecture is concerned with the design of computer systems. Regarding processor-based systems, it deals with the design and implementation of the interface between the software and hardware layers of such systems. We therefore discuss aspects of both layers and the interplay between them.
- Computer Architecture builds upon previous studies in digital circuits and systems design, assembly language, microprocessor operations, VHDL description, simulation and FPGA implementation.
- A study of Computer Architecture complements the study of languages and algorithms, compilers, and operating systems on the software side of the interface, as well as embedded processors, FPGAs and VLSI design on the hardware side of the interface.

Student Learning Outcomes

After completing this course, students will:

- know how computing systems are structured with a focus on the processor, cache, memory and system buses;
- understand single-cycle, pipelined, uni- and multri-processor models;
- understand why computers are structured the way they are;
- understand the importance of maximising performance and minimising power as drivers for engineering enhancement in computer systems design;
- have developed skills in designing architectural solutions and describing designs using VHDL;
- have become acquainted with the use of simulation for validating designs and studying architectural enhancement;
- have developed skills in obtaining and describing simulation results; and
- have practiced professional skills including project management, results analysis, critical discussion, structuring of information, oral and written presentation, and report writing.

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

Graduate Capability	Acquired in
scholarship: understanding of their discipline in its interdisciplinary context	Lecture content
scholarship: capable of independent and collaborative enquiry	Team project
scholarship: rigorous in their analysis, critique, and reflection	Project assessment
scholarship: able to apply their knowledge and skills to solving problems	Team project
scholarship: ethical practitioners	Project implementation
scholarship: capable of effective communication	Presentations and project report
leadership: enterprising, innovative and creative	Team project
leadership: capable of initiating as well as embracing change	Project enhancement
leadership: collaborative team workers	Team project
professionalism: capable of independent, self-directed practice	Tutorial problems
professionalism: capable of lifelong learning	Lecture content
professionalism: capable of operating within an agreed Code of Practice	Project implementation
global citizens: culturally aware and capable of respecting diversity and acting in socially just/responsible ways	Team project

Assumed Knowledge

Before commencing this course, students should:

- be familiar with a procedural programming language such as C;
- have experience programming in an assembly language;
- understand the principles of digital design;
- have experience using HDLs and CAD tools to specify and simulate digital components and simple systems; and
- have some experience implementing digital designs in FPGA devices.

These are assumed to have been acquired in the 1st and 2nd year core courses of our Computer Engineering degree, or equivalent:

- COMP1917 Computing 1
- COMP1927 Computing 2
- COMP2121 Microprocessors and Interfacing
- COMP3222 Digital Circuits and Systems

Teaching Rationale

Our teaching rationale is founded on the following ideas:

- We have a lot to cover in engineering study: understanding scientific theory, learning how to apply theory to design, developing designs, implementing and testing designs, creating products, and gaining an appreciation for professional issues; we want to try to integrate many of these as you progress to 3rd & 4th year.
- Provide theoretical foundation for knowledge:
    - Develop theory from observations of natural phenomena
    - Apply methodical approach to obtaining, recording, classifying and analysing observations
    - Instill a sense of the intrinsic value of the scientific method
    - We therefore encourage enquiry and especially questions in class
- Develop skills in design and critical analysis/evaluation of design options. These need to be gained over time from being exposed to lots of good and bad designs, from reflecting upon and discussing them, and most importantly, from practicing and learning from mistakes.
- We want to emphasize the need to carry out a thorough design prior to implementation.
- We would like to provide you with opportunities to practice and develop your professional skills, particularly those that relate to the presentation of ideas.

Teaching Strategies

- The course consists of lectures, tutorials, laboratory sessions and a sequence of assessable projects. You will need to work outside of formal hours to complete your project and to study and reflect upon the course material.
- Lectures will be used for presenting the background material, explaining the motivation behind computer designs, drawing comparisons between competing architectures, connecting material with prior knowledge, illustrating avenues for extension to the presented material, and guiding the direction of the course.
- Relevant tutorial questions will be posted on a weekly basis to the course website; their solution will be discussed during tutorial classes.
- Labs provide a venue for developing and exercising design, description, and simulation skills. The labs are intended to pave the way for the design and implementation of the design projects. They are also the primary venue for group meetings and work. You will be expected to build on provided code and gain proficiency at gathering and interpreting simulation results.
- In this course you will work on a major project that contains both individual and group work components. You will need to present your design during your tute & lab classes and to submit a final project report. Tute & lab classes will be used for project presentations and evaluation. All lab participants will be engaged in assessing the project work and presentations of others.

Assessment

Assessment in COMP3211/9211 will be based upon the following components:

- A 50 minute Quiz in the lecture class on Wednesday, 18 April (Week 7), worth 20% of your final mark and covering materials addressed in the first **6 (six)** weeks;
- A two-hour Exam worth 45% of your final mark at the conclusion of the course on all materials. You must obtain at least 40% of the available marks in the final exam in order to pass the course; and
- Project work and progressive assessment worth 35% of the final mark in this course.

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

No schedule is available (yet)

Resources for Students

Text:

- Computer Organization and Design: The Hardware/Software Interface, D.A. Patterson and J.L. Hennessy, 4th

Ed., Morgan Kaufmann, 2009.

Computer Architecture References:

- Computer Architecture: A Quantitative approach, J.L. Hennessy and D.A. Patterson, 3rd Ed., Morgan Kaufmann, 2003.
- Structured Computer Organisation, A.S. Tanenbaum, 5th Ed., Prentice-Hall, 2006.
- Computer Organization & Architecture: Designing for Performance, W. Stallings, 6th Ed., Prentice-Hall, 2003.
- Computer Organization, C. Hamacher, Z. Vranesic, and S. Zaky, 5th Ed., McGraw Hill, 2002. o Digital Design: Principles and Practices, J.F. Wakerly, 3rd Ed., Prentice-Hall, 2001.

VHDL References:

- VHDL Starter's Guide, S. Yalamanchili, 2nd Ed., Prentice Hall, 2005.
- A VHDL Primer, J. Bhasker, 3rd Ed., Prentice Hall, 1999.
- The Designer's Guide to VHDL, P.J. Ashenden, 2nd Ed., Morgan Kaufmann, 2001.
- Numerous on-line resources...

Course Evaluation and Development

Your feedback is welcome at anytime. Simply email or speak to a member of the course staff.

This course is also evaluated each session using the CATEI system.

In the previous offering of this course, students noted that the project work was interesting and tied in well with the lectures, but was not well integrated with the lab work.

Based on their comments, this year we aim to align the lab work more closely with the project requirements.