Faculty of Engineering and the Built Environment

Department of Electrical Engineering

Course handout : EEE4084F 2014

Course Name:	EEE4084F DIGITAL SYSTEMS
SAQA Credits:	20
Pre-requisites:	CSC4015Z (Operating Systems II), EEE3064W or EEE3017W (>70%)
Co-requisites:	None

Course convenor:	Simon Winberg
Email address:	simon.winberg@uct.ac.za
Office location:	6.13 Menzies Building
Consultation hours:	09h30-10h30 Monday; 10h30-11h30 Thursday. Alternatively email to request consultation.
Course lecturer:	Dr. Simon Winberg
Teaching assistants:	Mr. Lerato Mohapi

Lecture venue:	Chem Eng Seminar Room (Tue 9am), LCOM 2G (Tue 3pm), Menzies 11 (Thurs 2pm)
Lecture days and time:	Tuesday 9am-10am, 3pm-4pm, Thursday 2pm-4pm
Tutorial venues	Blue Lab
Tutorial day and time	Monday 9h00 to 11h00, Wednesday 9h00 to 11h00
Intensive block	N/A

Course objectives

The objective of this course is to develop an understanding of the concepts involved in the design and development of high performance and special-purpose digital computing systems, in terms of both hardware and software design. The course builds on a basic understanding of parallel computing principles, expanding the students' expertise into the specialized fields of reconfigurable computing and high performance parallel computing. Aspects related to Very Large Scale Integration (VLSI) and the application of Hardware Description Languages (HDL) to special purpose System on Chip (SoC) design are studied but due to time and laboratory restrictions VLSI in particular is covered only briefly, having the focus on chip design related activities restricted to theoretical study and functionality prototyping using FPGAs (as opposed to delving into more detailed real-world fabrication and packaging issues). The coursework involves five laboratory assignments and two mini projects: the practicals are designed around supporting the projects, in particular the first two practicals are intended to foster insights that lead into the conceptual assignment (first mini project) and the last two practical helps students to get up-to-speed on FPGA-based prototyping and application development that facilitates their practical investigation for the FPGA SoC / application accelerator prototyping that constitutes a significant portion of the larger second project.

A number of compulsory quizzes are held, for which advanced notice and a syllabus is always provided in good time prior to the quiz date. All the quizzes / tests / exams are closed book unless explicitly announced beforehand. The lecture sessions include presentations by lecturers, seminars and workshops during which students learn fundamental theories, brainstorm ideas, and discuss influential and recent publications in the field. The lecturer tried to get in a industry partner or guest lecturer to present on relevant state-of-the-art technology or projects.

Note: Due to on-going curriculum changes, in 2014 this course is specified as core according to the UCT EBE undergraduate courses handbook and it will have Engineering Council of South Africa (ECSA) Exit Level Outcomes (ELOs) attached which will be assessed during certain assignments and tests (as described in the ELO assessment strategy below). ELO 1 and ELO 2 will both be attached to EEE4084F. During this transitional stage, some ECE students may be taking other courses that have both these ELOs attached as well; regardless of whether that is the case or not, students are recommended to ensure they satisfactorily complete and pass the relevant assignments linked to these ELOs (and where necessary second attempts in the even the first attempt is not successful). Complete all the ELOs (including ELO1 and 2 attached to this course) are a necessary requirement for the degree programme; students will only be granted the degree once having completed all ten ECSA ELOs.

More information about EEE4084F is available at the public website: http://www.rrsg.ee.uct.ac.za/courses/EEE4084F/

Exit Level Learning Outcomes

		Only those cells shaded below are at exit level.									
Students successfully completing this course will have the following:	At Exit level	ELO 1	ELO 2	ELO 3	ELO 4	ELO 5	ELO 6	ELO 7	ELO 8	ELO 9	EL10
A. Knowledge (Information plus Understanding)											
1. Parallel computing fundamentals, concept of automatic parallelism; undestand the difference between microprocessor-based computer systems, embedded computer systems and reconfigurable computers. Concepts of temporal and spatial programming. Current trend in high performance embedded computing (HPEC).	N									8	
2. Patterns in parallel computing (seminar 1) and in HPEC (seminar 2) design. Knowledgeable of the steps in designing parallel programs (these steps are generic to both microprocessor- and FPGA-based/embedded parallel systems). Understanding of parallel design patterns.	N	8								8	
3. Theories of benchmarking digital systems (incl. speed and power benchmarking) Parallel programming models (data parallel model; message passing model, etc.) Understanding of the common parallel architecture models (including Von Neumann architecture, Flynn's taxonomy, etc.), memory access architecture models. Can draw on knowledge of case studies to illustrate some of these architecture models.	N	8	8							8	
4. Knowledge of computer subsystems, including DMA, bus management systems, memory types and memory management systems and interrupts among others. Fundamentals of microprocessor-based and FPGA-based reconfigurable computing. Knowledge of the fundamental reconfigurable computing building blocks. Understand theory and limitations of using an automatic C → HDL conversion tool to translate a specialized form of C code to HDL.	N	8	8							8	
B. Skills (Application of Knowledge)											
1. Can develop Pthread programs in C or C++, able to compile and run this on a digital standard computer architecture. Able to translate a complex sequential algorithm into a parallel version, and can implement and run the resultant algorithm on a computer cluster using OpenMP. Using a hybrid computer, comprising a multicore standard architecture with and Programming Graphic Processor Unit (GPU) running highly parallel code. (includes using CUDA). Can design and code FPGA-based firmware for use on a reconfigurable computer platform (i.e. useful for prototyping a custom HPEC system).	N	8	8								
2. Benchmarking of parallel computer systems (microprocessor-based systems and systems making use of GPUs). Benchmarking embedded high performance computing systems (both FPGA-based computer systems and microprocessor computer systems).	N	8	8								
3. Ability to describe HPEC design ideas. Can describe (in writing with figures) how to go about designing a HPEC system, or HPEC subsystems, that solves an open-ended application development problem.	Y	**8**	**8**	8							
4. Able to validate and verify digital system design. Can use correlation to compare results produced by different platforms. Able to do performance analyses on the design of HPEC systems including analysis of communication performance between computer systems. Given a detailed design of a HPEC system, the student can perform paper-based calculations to estimate the performance of the HPEC system. Using this data to identify performance bottlenecks, areas to optimise design costs, amongst suggesting possible other improvement to the given computer design.	Y		**8**								
5. Able to understand and describe the principle characteristics of a digital system, its overall design process, design software and hardware for such systems. Able to lead a technical seminar on such a topic.	N			8			8				
C. Values and Attitudes											
1. Appreciation of digital system / HPEC design to support real time applications.	N							8			
2. Stimulate an interest in this branch of Electrical Engineering.	N							8			8
3. Able to contend with open-ended problems requiring thinking and intuition.	N										
4. Appreciate the need of teamwork in digital systems design and implementation.	N								8		

Detailed course content

The course is divided into to parts: Part A and Part B. Part A runs in the first term, and Part B runs in the second term; these are explained below:

Part A: Microprocessor-based parallel computing and theory with abstract study of special-purpose CPU/processor/SoC design. Three practicals related to parallel computing and 'cognitive assignment' mini-project concerning partial design and theoretical analysis of a special-purpose CPU/SoC.

Part B: Reconfigurable computing and the design and development of digital logic to implement digital accelerator (or special purpose digital SoC systems) running on FPGA platforms (these digital accelerators could also be considered as a means to prototype functionality aspects of digital SoC system, albeit at a much larger scale, and slower performance, that what can be achieved on a silicon wafer).

Seminars are run throughout the semester. The students are divided into Seminar Facilitation Groups (SFGs), where each group has a turn to present a reading and to lead the seminar on the topic (the first seminar is done by the lecture as a means to show the students what is expected during the seminars – the lecturer grades the groups' performance and their understanding of their seminar material which counts for marks). All students are expected to attend the seminars; participation marks are awarded for attendance (these count a small portion of the coursework mark).

The lectures include fundamental theories, design practices and techniques related to the design of digital systems. The lectures cover a number of case studies related to real systems that were developed. The lectures introduce theories and techniques that connect with and help students to get started on the laboratory practicals and project work.

During the first term (Part A) students focus on fundamentals of parallel computing architectures and parallel programming. During this term students focuse on laboratory pracs and homework tasks. Mid-way in term 1 the students start on the conceptual assignment related to the design and theoretical assessment of a special-purpose CPU or SoC system. This conceptual exercise, as the name suggests, is largely a thinking exercise and high-level design task; it involves writing a report that is due at the end of the first term.

During the second term (Part B) the course has an emphasis on the use of field programmable gate arrays (FPGAs) and HDL programming in relation to design and application development for ReConfigurable (RC) hardware platforms. A RC platform is utilized in the pracs and project as a means to accelerate computation. Students' prior experiences in HDL coding, hardware/software peripheral interfacing and other more embedded software related development issues are built upon to development an HDL hardware accelerator component or specialized peripheral, together with necessary PC/host computer software development, to prototype a (fairly simple) hardware accelerator or functionality limited SoC. This larger project is called the 'Your Own Digital Accelerator' (YODA) Project, and is worked on by a team of two or three students. This purpose of this fairly substantial project is to integrate the students prior experiences (of embedded systems development etc.) together with the new techniques they learn in this course to build and test the YODA device, prototyped using an FPGA development where necessary with some elements simulated in software (i.e., aiming towards a hardware-in-the-loop type of approach where the system might run more in 'slow time' / simulation time as opposed to real-time). Besides HDL and software development and simulation, the project touches on issues of problem description, problem solving, design and implementation, hardware/software interfacing, communications protocol design, experimentation and culminates in a final acceptance testing demonstration.

Outline of course structure

The following structure is generally somewhat consistent between versions of this course, but the specific topics covered may change a bit each year.

Term 1	Term 2
Introduction to Digital Systems and Reconfigurable Computing - Microprocessor-based vs. FPGA-based solutions for RC - Focusing on application development for FPGA-based RC system - Intro to platform & tools we will use - UML (recap) and other modelling notations used in this course - Terminology (e.g. Golden Measure; Spatial vs. Temporal computing) - Performance benchmarking; Size, Weight and Power benchmarking Parallel programming and architectures - Parallel computing fundamentals - Automatic parallelism - Techniques. Selected topics on timing in C or other lanaguages. - Important terms (Contiguous, Partitioned, Interleaved, Interlaced) - Parallel programming models: Data parallel model; Message passing model; Shared memory model; Hybrid model - Processor architecture types - Memory access architectures Design of Parallel Programs - Understanding the problem & requirements elicitation - Partitioning, Decomposition & Granularity, Communications - Identify data dependencies - Synchronization - Load balancing, performance analysis & tuning Additional concerns for parallel architectures - Cost of communication - Latency, bandwidth, effective bandwidth (and related calculations) - Blocking/non-blocking; synch/asynch; scope of communications Cloud computing - virtualization and other key technology factors GPUs and special-purpose accelerators - GPUs (issues and benefits); Delving into details of CUDA - Important CUDA-related terms - Design of parallel programs using for GPUs/CUDA Leading in to Term 2 - Presentation of the YODA (Your Own Digital Accelerator) project - A look at some of the best past projects (short video) - YODA topics prepared by Lecturer and suggestions requests by research colleagues. - Call for proposals for innovative YODA projects - Overview of the process concerning the design, implementation and acceptance testing of your future YODA prototype.	FPGA systems and related architecture - Recap programmable logics, HDL and VHDL - Performance benchmarking techniques - Overview of the variety of FPGAs and tools. Latest Trends. - Theoretical speed calculation of a logic design. - Comparing FPGA-based execution time to CPU execution time. - Base core equivalent (BCE), related cost calc. & benchmarking - Relevance of Amdahl's law to FPGA-based RC systems - The FPGA Platform that we will use Introduction to Verilog HDL - Most students in ECE have encountered VHDL but there are advantages to knowing some Verilog as well, the lectures delve into the syntax and some simple examples using Verilog, the open-source Icarus Verilog simulator and Xilinx QSim. Leading on to FPGA app coding. FPGA and computer board architectures - Configuration architectures - Designs concerns & vignettes (changes somewhat each year) - Development tools and toolflows - FPGA Fundamentals; Licensing of tools & IP - Different types of FPGAs; differences between PALs, CPLDs; Physical structure of PALs vs CPLDs; parts for I/O & control - Types of logic, metalogic, libraries, Xilinx scenarios - Use of lookup tables, megablocks etc. The RC Design Process - Recap of ES design cycle and HW/SW co-design - The FPGA Design Cycle - Specialized Topics: Technology-independent optimization; Sourcing Intellectual Properties (reading assignment); Technology Mapping; Placement & Routing Advanced FPGA techniques for RC development - Back to the FPGAs. Softcore & uCLinux O/S running on an FPGA platform - Using cache prediction, other options avail on FPGA - Coarse-grained Reconfigurable Devices and Multi-FPGA Systems Practical algorithms / methods for RC application development FPGA Advanced Topics - Memory types, Digital logic modular design, - DMA. Interrupt controllers. When to use latches & flip flops Benchmarking and selected further topics - Benchamarking using DMIPS, Dhrystone, Whetstone, Coremark - C → HDL automatic conversion (HandleC notation, how to write a C-style representation of a digital logic circuit).

Knowledge areas

Mathematics	Basic Sciences	Engineering Sciences	Design and Synthesis	Complementary Studies
0	0	50	50	0

Learning environment

Multi-modal. Combination of laboratory, lecture venue and online. Students will be expected to engage with material in various forms in a variety of different ways, but have much flexibility in where and how this material is accessed.

Suggested time allocation ***(Note that this needs to be done according to Appendix B in ECSA document PE-61)***

Learning Activity	Time (hours)
Lectures (48 x 0.75h each = 36h) (12 of the lecture slots are allocated to seminars)	36
Assimilation time, homework & independent study (48 x 1.25h) (incl. reading scheduled seminar papers, preparing group seminar)	60
Laboratory Practicals	35
Other contact time (tests, design reviews, present YODA)	8
Assignments (Wiki, Conceptual assignment, YODA project)	61
Total learning time	**200**

General assessment strategy

Assessment Task	Weighting	The following DP rules apply:
Tutorials, Laboratory Projects Tests Written Examination Hands-on Computer Exam Other (participation, essay/blog) **Total**	10% 20% 20% 40% 10% 100%	• Minimum 40% overall class average to write the final exam • Successful completion of both ELO1 and ELO2 (see details below)

Breakdown of assessment strategy for ELOs

Main outcome	Activities, skills or abilities contributing to the main outcome/objective	Alignment of assessment with relevant ELO descriptions
Learning outcome 1: Problem solving Identify, formulate, analyse and solve complex engineering problems creatively and innovatively.	Level Descriptor: Complex Engineering Problems:	The first attempt of ELO1 is in the form of the first term conceptual assignment.
	a) require in-depth fundamental and specialized engineering knowledge;	This task involves high-level design of a special-purpose CPU / SoC. Students are expected to draw on their learning from previous years, such a microcontroller programming courses and in particular EEE3064W / EEE3017W, to prepare the specification and (partial) design (but not implementation) for the CPU/SoC using combinational digital logic. (The brief for the general requirements of the CPU/SoC changes each year).
	and have one or more of the characteristics	
	b) are ill-posed, under- or overspecified, or require identification and refinement;	The student needs to identify what additional requirements and other information is needed in order to refine the specification of the CPU/SoC. The student needs to not only indicate/reference where such information needs to be obtained but needs to provide a brief literature review elaborating on some of the specific additional requirements chosen to improve upon the project requirements. (Clearly, the CPU/SoC considered is a very much simplified device compared to what would be expected for a commercial/real product.)
	c) are high-level problems including component parts or sub-problems;	Student needs to recommend a partitioning of the design space, consider interconnection busses, processing subcomponents (e.g. ALU, MU, etc.) that would form

		part of the proposed system. In situations the student is influenced by academic papers or datasheets for existing products, these sources need to be properly referenced. Furthermore, in situations where the development team meets with a postgraduate or one of an industry partners due acknowledgement of ideas and possible solutions, and if relevant reference, should be given appropriately.
	d) are unfamiliar or involve infrequently encountered issues;	The requirements of the CPU/SoC are specially planned by the lecturer to be somewhat unusual (indeed some of the proposed systems, such as a SIMD processor, have been academic thought experiments at best). Although the application of some of these product concepts are debatable they sure to comprise many unfamiliar design considerations that the students would not likely have encountered previously in other courses.
	and their solution have one or more of the characteristics:	
	e) are not obvious, require originality or analysis based on fundamentals;	Students need to prepare theory-based model for the proposed CPU/SoC to provide estimated characteristics, in terms of: performance (e.g., MOPS), size (PCB footprint and LE cost in mm^2), power (e.g. Watts used at max. clock speed for the most demanding OPs), etc.
	f) Formulates and presents the solution in an appropriate form.	Students need to prepare a form written report suitably structured to clearly but concisely convey the specification, high-level design and theoretical performance results for the conceptual assignment.
	g) require information from variety of sources that is complex, abstract or incomplete;	Student is expected to perform a (constrained) literature review and engage in meeting with the lecturer/their project manager (likely one of the postgrad tutors) to refine the specifications for the CPU/SoC device.
	h) involve wide-ranging or conflicting issues: technical, engineering and interested or affected parties.	The CPU/SoC conceptual assignment incorporates a variety of wide-ranging (but generally not conflicting) issues of a technical nature commonly encountered in real-world computing engineering projects.
Learning outcome 2: Application of scientific and engineering knowledge Apply knowledge of mathematics, natural sciences, engineering fundamentals and an engineering specialty to solve complex engineering problems.	Level descriptor: Knowledge of mathematics, natural sciences and engineering sciences is characterized by:	The first attempt of ELO2 is in the form of the first term conceptual assignment.
	* A systematic, theory-based understanding of the natural sciences applicable to the discipline;	During the conceptual project, a high-level design problem is presented to the student. This design problem is 'ill defined' as in the requirements and specifications for the system are only loosely and in parts rather vaguely described. The student is required to engage in a variety of problem-solving tasks, which draw on their mathematical and scientific engineering knowledge, in order to refine the requirements and specifications, leading towards a more complete description of the proposed system. Furthermore, once a proposed system design has been established, the student needs to carry out a paper-based theoretical performance analysis of the design giving research-based approximations for the size, power and performance characteristics of the system. The students should demonstrate in their report for this assignment an appropriate understanding of relevant discipline-specific theories.
	* Conceptually-based mathematics, numerical analysis, statistics and formal aspects of computer and information science to support analysis and modelling applicable to the discipline;	As part of the design work and performance calculations, students will be expected to draw on relevant mathematics, numerical analyses (e.g., using OCTAVE or MATLAB) to propose an effective CPU/SoC design and performance analysis results. Appropriate use of visualization and technical writing methods are expected to be provided in the report.
	* A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline;	The report for the Conceptual Assignment includes a section titled 'Proposed Development Processes'. For this section, the student is expected to provide a brief overview of the way by which development team would be structured and major design phases for the CPU/SoC conceptual assignment were this project to be carried out as a real-world industry-based design project.

	* engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.	The student is expected, using the lectures and literature as guidelines, to develop and report on their own performance analysis methodology used to establish the theoretical performance results for the CPU/SoC system. In addition they are expected to build on what they have learned in order to report on the 'Proposed Development Processes' that they would recommend be followed if the conceptual CPU/SoC system became a real project.
	Mathematics, natural science and engineering sciences are applied in formal analysis and modelling of engineering situations, and for reasoning about and conceptualizing engineering problems.	

Information specific to ELOs

Exit Level Outcome 1
Where and how is this learning outcome assessed? In Term 1, students are required to complete the Conceptual Assignment – this constitutes the first of two projects for this course. This first project is so named as it does not involve implementation work (fabricating the proposed special-purpose CPU/SoC that is the topic of the project is out of the scope and available resources for this course). Accordingly, this project follows a largely theory-based approach. The students are given incomplete requirements and brief specification-related suggestions. The student is then expected to elicit further requirements and build upon these in order to develop more refined requirements and specifications for the proposed CPU/SoC. A high level system design then needs to be developed followed by a paper-based and theory-guided performance analysis of the system. A specifications review will be conducted, which will be run by the lecturer/tutor/TA (if not the lecturer then a postgraduate with a suitable understanding of these issues). This specifications review is intended as a formative assessment task, essentially providing an early warning to the student for a likely of possibly not completing the report satisfactorily. The main form of assessment for this ELO is the conceptual assignment report, which is a comprehensive technical document that is expected to respond to all the tasks defined in the conceptual assignment, as explained in above section for "Breakdown of assessment strategy for ELOs".
What constitutes satisfactory performance? The report must be a professional technical document that provides adequate responses to all the assignment tasks as defined in the conceptual assignment handout. The proposed system design must be sufficiently thorough and of a quality approved by the lecturer supported as needed by an appropriately skilled TA/collaborator (a physical implementation and testing of the proposed system is beyond the scope of this course; therefore the assessment will be carried out as a theory-based approach with the evidence scrutinize as a partly tacit approach relying on the examiner's experience and expert knowledge).
What strategy is to be followed should this learning outcome not be satisfactorily attained? The student will be required to rework the relevant parts of the design / performance analysis calculations and resubmit the report. **Note:** The mark for the first submission of the report will be recorded in the course mark so as to make things fair for the rest of the class (and to discourage students from being sloppy for the first submission). Students should be conscious of the need to have a 40% minimum coursework mark in order to obtain a DP; so getting ELO1 requirements passed after a report resubmit does not guarantee that the DP requirement for the course will be satisfied.
Exit Level Outcome 2
Where and how is this learning outcome assessed? This ELO is assessed based on the report for the conceptual assignment for this course. This report will be used to establish the student's competence in applying theories (and relevant knowledge of mathematics, natural sciences and engineering fundamentals) to satisfactorily prepare a comprehensive system design proposal and theoretical performance analysis for the complex engineering problems as described in the conceptual assignment project for this course. Students will be expected to use appropriate numerical computations (e.g. using OCTAVE / MATLAB) to support or supplement the evidence supporting their proposed system design and performance analysis.
What constitutes satisfactory performance? The relevant models used to express the system design for the conceptual CPU/SoC, and the theoretical performance analysis of the system based on these models, needs to be technically correct and suitably supported by motivation related to the student's design/model and needs to provide well explained calculations/formulae used to establish the performance results. These calculations need to be at suitable level of accuracy with the units (e.g. microseconds) clearly indicated (very course grained guesses as to the performance estimates and operation completion things that inadequate support the calculations will not be acceptable and may require the report to be resubmitted). The report will be interrogated through expert review by the lecturer and if needed with support from a TA/collaborator to assess the technical accuracy and quality of the report content.
What strategy is to be followed should this learning outcome not be satisfactorily attained? The student will be required to rework the relevant parts of the report / performance analysis and resubmit the report. **Note:** The mark for the first submission of the report will be recorded in the course mark so as to make things fair for the rest of the class (and to discourage students from being sloppy for the first submission). Students should be conscious of the need to have a 40% minimum coursework mark in order to obtain a DP; so getting ELO2 requirements passed after a report resubmit does not guarantee that the DP requirement for the course will be satisfied.

Prescribed Books/Reading Materials/Notes

- Text book: Martinez, Bond and Vai. *High Performance Embedded Computing Handbook*. New York: CRC Press. 2008
- Journal articles (available on VULA / linked to on the OpenUCT course website)
- Online course notes (available on VULA)
- Web sites:
  - Public / OpenUCT website: http://www.rrsg.ee.uct.ac.za/courses/EEE4084F
  - Vula site: See EEE4084F in your Vula tabs

Absence: The continuous assessment marks will be adjusted to allow for absence only on the following grounds:

- A medical certificate for absence of 3 or more consecutive days
- Death of an immediate family member (parent or sibling)
- Pre-arranged absence to represent a University, provincial or national team.

Academic dishonesty: Plagiarism is a very serious offence and usually leads to disciplinary action that could include expulsion from the university. Therefore, recognise the work of others in any submission. Details of referencing methods are widely available on the Web.