University of Dublin
Trinity College
Faculty of Engineering & System Sciences

BAI Computer Engineering (Stream D)
BAI Electronic & Computer Engineering (Stream CD)
BAI Electronic Engineering (Stream C)

Course Handbook

Mike Brady & Hitesh Tewari

November 21, 2005
Contents

1 Introduction 7
  1.1 Administration ................................................. 9
  1.2 Scope of This Document ......................................... 9
  1.3 Disclaimer ....................................................... 9
  1.4 Aims and philosophy of the course ........................... 10
  1.5 Course Duration and Structure ................................. 10
    1.5.1 Junior Sophister ......................................... 10
    1.5.2 Senior Sophister Year ................................. 12

2 Junior Sophister Year 13
  2.1 Overall Structure ............................................. 13
    2.1.1 Core Courses ............................................ 13
    2.1.2 Specialist Courses ...................................... 13
    2.1.3 Project Design and Implementation .................... 14
    2.1.4 Laboratories ............................................ 14
    2.1.5 Overview ................................................ 15
  2.2 Timetables ..................................................... 18
  2.3 Lab Groups and Timetable ..................................... 20
  2.4 Course Details ................................................ 21
    2.4.1 Signals and Systems (3C1) ............................. 21
    2.4.2 Electronic Engineering I (3C2) ....................... 24
    2.4.3 Electronic Engineering II (3C3) ....................... 27
    2.4.4 Electromagnetism and Optoelectronics (3c4) ......... 30
    2.4.5 Telecommunications (3C5) .............................. 31
    2.4.6 Electronic Systems Design & Implementation Project
           (3C6) .................................................. 35
2.4.7 Microprocessor Systems 1 (3D1) ........................................ 40
2.4.8 Microprocessor Systems 2 (3D2) ....................................... 44
2.4.9 Concurrent Systems (3D3) .............................................. 47
2.4.10 Computer Aided Design 1 (3D4) .................................... 50
2.4.11 Software Design and Implementation (3D5) ...................... 54

2.5 Laboratory and Project Design and Implementation Details .... 57
2.5.1 Programme ............................................................... 58
2.5.2 Timetable ................................................................. 59
2.5.3 Assessment ............................................................... 60
2.5.4 Rules ......................................................................... 60

3 Senior Sophister Year ...................................................... 63
3.1 Overall Structure .......................................................... 63
3.1.1 Specialist Courses ...................................................... 63
3.2 Timetable ................................................................. 64
3.3 Course Details ............................................................ 66
3.3.1 Management for Engineers (4E1) .................................. 66
3.3.2 Final Year Project (4E2) .............................................. 72
3.3.3 Microelectronic Circuits 1/2 (4C2/4C7) .......................... 75
3.3.4 Telecomms / Digital Communications (4C4/4C10) ............ 78
3.3.5 Digital Signal Processing / Digital Control Systems (4C5/4C3) .... 81
3.3.6 4C6/4C11 Microelectronic Technology and Optoelectronics ........................................... 85
3.3.7 Integrated Systems Design (4S1) ................................. 88
3.3.8 Networking and Advanced Microprocessor Systems (4D1) .... 92
3.3.9 Knowledge and Data Engineering (4D2) ......................... 98
3.3.10 Operating Systems and Distributed Systems (4D3) ........... 100
3.3.11 Computer Vision and Robotics (4D4 Section B) ............... 104

4 Further Information ....................................................... 105
4.1 Examiners Rules 2005/2006 ............................................. 105
4.2 Freshman and Junior Sophister Engineering 2005/06 ............. 105
4.2.1 Overall Grading ......................................................... 105
4.2.2 Subject Results .......................................................... 107
4.3 Senior Sophister Engineering 2005/06 ................................ 107
4.3.1 Classes of degree ....................................................... 107
4.3.2 Contribution from Junior Sophister year ......................... 108
Chapter 1

Introduction

Welcome to the sophister specialisations Electronic Engineering, Electronic and Computer Engineering and Computer Engineering. As you’ll know by now, these are referred to a Stream C, Stream CD and Stream D respectively. To put the streams in context, some background: The Faculty of Engineering and Systems Science offers a four year Engineering Science degree course, leading to the award of the BAI—the *Baccalaureus in Arte Ingeniaria*.\(^1\) The BAI degree has engineering science at its core, and the first two years of the course—the *freshman years*—provide a broad foundation in mathematics, the physical sciences and the fundamentals of computing, mechanical engineering, electronic engineering and civil engineering. Students then specialise in one of five streams in the sophister years:

- Civil, Structural and Environmental Engineering (Stream A)
- Mechanical and Manufacturing Engineering (Stream B)
- Electronic Engineering (Stream C)
- Electronic and Computer Engineering (Stream CD)
- Computer Engineering (Stream D)

\(^1\)For historical reasons, engineers in Trinity are also awarded the ordinary degree of BA; the full set is BA, BAI
The streams are organised around a core of engineering science subjects—mathematics, management, the engineer in society, optional courses in French or German and, in final year, a substantial project. The specialist subjects are offered by the engineering departments. The Department of Civil, Structural and Environmental Engineering offers stream A, stream B is offered by the Department of Mechanical and Manufacturing Engineering. The Departments of Computer Science (CS) and Electronic and Electrical Engineering (EEE) offer the last mentioned three streams: Electronic Engineering (Stream C), Electronic and Computer Engineering (Stream CD) and Computer Engineering (Stream D).

The objective of the BAI degree offered by Departments of Electronic and Electrical Engineering and Computer Science is to produce well-rounded
graduates having a strong grounding in analytical skills and flexibility to adapt to the advances in electronics technology, computer systems and communications systems.

1.1 Administration

The BAI degree, of which these streams are part, is organised and administered by the Schoole of Engineering assisted by departmental coordinators.

- The Dean of the Faculty of Engineering and Systems Sciences is Dr Brian Foley, email: Brian.Foley@tcd.ie.
- The School of Engineering Administrator is Michael Slevin, email: maslevin@tcd.ie, telephone 1796.
- The coordinator for the Department of Computer Science is Dr Hitesh Tewari, email: htewari@cs.tcd.ie.
- The coordinators for the Department of Electronic and Electrical Engineering are, for the Junior Sophister Year: Professor William Coffey, email: wcoffey@mee.tcd.ie, and for the Senior Sophister Year: Professor Paul Fannin, email: pfannin@tcd.ie.

1.2 Scope of This Document

This is a handbook for Junior and Senior Sophister BAI students of Electronic Engineering (Stream C), Electronic & Computer Engineering (Stream CD) and Computer Engineering (Stream). It deals with the Junior and Senior Sophister years of the C, CD and D streams exclusively. For information on the Freshman years of the BAI and for information on the other streams of the BAI, the reader is referred to the BAI website, at http://www.tcd.ie/engineering/Courses/BAI/.

1.3 Disclaimer

The definitive source for all information about the university and about the BAI is the University of Dublin Calendar, available at
http://www.tcd.ie/Secretary/College_Calendar/. If there is a divergence between the Calendar and this handbook on any matter, the Calendar is authoritative. If you notice any such divergence, please bring it to the attention of one of the coordinators.

1.4 Aims and philosophy of the course

This four year full-time degree course is designed to ensure a sound understanding of scientific and engineering principles and their application to the solution of practical problems in Electronic and Computer Engineering. Comprehensive studies in Mathematics, Physical Sciences, Computer Science and Engineering Science in the first two years of the course provide a firm foundation for advanced work in Electronic and Computer Engineering subjects in the two senior years.

1.5 Course Duration and Structure

The BAI course is a four-year full-time programme based on a 24-week academic year.

1.5.1 Junior Sophister

The Junior Sophister year is organised in two semesters (half-years) arranged to overlay the traditional Trinity three-term structure. Semester 1 runs from the start of the Michaelmas term to the end of the second week of Hilary term. A two-week gap follows, during which the Junior Sophister engineering part 1 examinations take place. (Three subjects are examined at this time: 3C1, 3C2 and 3D1). This is followed by Semester 2, running from the fourth week of Hilary term to the end of the Trinity term. This is followed by the Annual examinations period.
The course comprises three core subjects, viz. \textit{3E1 Engineering Mathematics V}, \textit{3E2 Engineering Mathematics VI} and \textit{3E3 The Engineer, Management and Society}, an optional language module in French or German, six specialist subjects and two terms of project design and implementation. Specialist subject courses are provided by each department (3C\textit{x} by EEE and 3D\textit{x} by CS) and last for one semester.

\textbf{Laboratories}

Laboratories are associated directly with each individual CS subject. A programme of Electronic Engineering laboratories, associated with all EEE subjects, is centrally organised within the EEE department.

A weekly three-hour laboratory is scheduled for project design and implementation in the Michaelmas and Hilary terms.
Examination Periods

Three subjects are examined during the Junior Sophister engineering part 1 examinations period at the end of the first semester, i.e. around the end of January. The other examinations are held in the Annual examinations period.

1.5.2 Senior Sophister Year

In the Senior Sophister year, lectures are held in the Michaelmas and Hilary terms; no lectures are scheduled for the Trinity term.

The course comprises two core subjects, viz. 4E1 The Engineer, Management and Society, 4E2 Engineering Project, and four specialist subjects (pairs of subjects in the case of EEE) provided by the departments.

Laboratories

Laboratories are associated directly with each individual subject as determined by the lecturers.

Examination Periods

All Senior Sophister examinations are held in the Annual examinations period.
Chapter 2

Junior Sophister Year

2.1 Overall Structure

The overall structure of the streams C, CD and D is as follows: three core subjects—3E1 Engineering Mathematics V, 3E2 Engineering Mathematics VI and 3E3 Engineer, Management and Society—and an optional language module in French or German are complemented by six specialist subjects and two terms of project design and implementation. Core subjects run through both semesters and specialist courses are one semester long.

2.1.1 Core Courses

The core courses have two lectures per week in both semesters. In addition, 3E1 and 3E2 have one tutorial per week. French for Applied Scientists and German likewise has two lectures per week in both semesters.

2.1.2 Specialist Courses

Specialist courses generally comprise three lectures and one tutorial per week for one semester. Courses originated by the CS department courses also have practicals associated with them. In the first semester, the specialist courses are 3C1 Signals and Systems, 3C2 Electronic Engineering 1 and 3D1 Microprocessor Systems 1, whichever stream you choose. Examinations in these subjects are held in the Junior Sophister engineering examinations part 1 examination period, which is in the second week of the two-week gap between semesters.
Specialist Courses by Stream

The selection of specialist subjects in the second semester is different for each stream. For the C stream, the second-semester subjects are 3C3, 3C4 and 3C5. For the CD stream, the second-semester subjects are 3D2, 3D3 and 3C5. For the D stream, the second-semester subjects are 3D2, 3D3 and 3D4.

2.1.3 Project Design and Implementation

Separate design and implementation projects are organised for the Michaelmas and Hilary terms by each department. These are coded 3C6A and 3C6B (Electronic & Electrical Engineering) and 3D5A and 3D5B (Computer Science Department).

Project Design and Implementation by Stream

Students in the C stream take both design projects from the EEE department, and so undertake 3C6A in the Michaelmas term and 3C6B in the Hilary term. Students in the CD stream take a design project from each department, undertaking 3D5A in the Michaelmas term and 3C6B in the Hilary term. Students in the D stream take both design projects from the CS department, and so undertake 3D5A in the Michaelmas term and 3D5B in the Hilary term.

Session Times

The project design and implementation sessions run in the Michaelmas and Hilary terms on Friday afternoon for three hours. In addition, a one hour orientation session in 3D5 is held on Thursday mornings.

2.1.4 Laboratories

Laboratories associated with the EEE department are held on Tuesday afternoons for three hours. Laboratories associated with 3D1 and 3D3 are held on Wednesday afternoons. Laboratories associated with 3D2 are held on Friday afternoons in the Trinity term for three hours. Laboratories associated with 3D4 are held on Thursday mornings. See page 57 for more details.
2.1.5 Overview

The structure of each stream is shown in the next three diagrams.

![Diagram of Junior Sophister Computer Engineering (D Stream) Structure](image)

Figure 2.1: Junior Sophister Computer Engineering (D Stream) Structure
### Figure 2.2: Junior Sophister Electronic & Computer Engineering (CD Stream) Structure
### 2.1. OVERALL STRUCTURE

#### Michaelmas Term
- 3E1 Engineering Mathematics V
- 3E2 Engineering Mathematics VI
- 3E3 The Engineer, Management and Society
- French or German [Optional]
- 3C6A Design Project
- 3C1 Signals & Systems
- 3C2 Electronic Engineering 1
- 3D1 Microprocessor Systems 1

#### Hilary Term
- 3C6B Design Project
- 3C5 Telecommunications
- 3C4 Electromagnetism & Optoelectronics
- 3C3 Electronic Engineering 2

#### Trinity Term

<table>
<thead>
<tr>
<th>Semester 1</th>
<th>Semester 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.3: Junior Sophister Electronic Engineering (C Stream) Structure
2.2 Timetables

The official timetable is published on the notice board in the museum building. Up-to-date timetables for each semester, including extra details of laboratory arrangements, are maintained on the web at http://www.tcd.ie/engineering/Courses/BAI/Joint_CS_EE/JS/Timetables/semester1.html and http://www.tcd.ie/engineering/Courses/BAI/Joint_CS_EE/JS/Timetables/semester2.html.

![Timetable for Semester 1](image)

Figure 2.4: Timetable for Semester 1

Legend: [MT only] means Michaelmas Term only, [HT only] means Hilary Term only
## 2.2. TIMETABLES

![Timetable for Semester 2](image)

**Figure 2.5: Timetable for Semester 2**

Legend: HT means Hilary Term, TT means Trinity Term
2.3  Lab Groups and Timetable

<table>
<thead>
<tr>
<th>Group (num people)</th>
<th>Oct. 11</th>
<th>Oct. 18</th>
<th>Oct. 25</th>
<th>Nov. 1</th>
<th>Nov. 8</th>
<th>Nov. 15</th>
<th>Nov. 22</th>
<th>Nov. 29</th>
<th>Dec. 6</th>
<th>Jan. 10</th>
<th>Jan. 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 (3)</td>
<td>F</td>
<td>A1</td>
<td>A1</td>
<td>A1</td>
<td>S1</td>
<td>S2</td>
<td>OP</td>
<td>T2</td>
<td>T1</td>
<td>T1</td>
<td>D1</td>
</tr>
<tr>
<td>C2 (4)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>A1</td>
<td>S1</td>
<td>S2</td>
<td>IC</td>
<td>OP</td>
<td>T1</td>
<td>T2</td>
<td>D1</td>
</tr>
<tr>
<td>C3 (4)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>A1</td>
<td>S1</td>
<td>S2</td>
<td>T2</td>
<td>IC</td>
<td>OP</td>
<td>T1</td>
<td>D1</td>
</tr>
<tr>
<td>CD1 (3)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>S1</td>
<td>S2</td>
<td>T1</td>
<td>D1</td>
<td>O2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD2 (4)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>S1</td>
<td>S2</td>
<td>T1</td>
<td>D1</td>
<td>D2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD3 (3)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>S1</td>
<td>S2</td>
<td>D1</td>
<td>T1</td>
<td>D2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD4 (4)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>S1</td>
<td>S2</td>
<td>D1</td>
<td>T1</td>
<td>D2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1 (1)</td>
<td>F</td>
<td>A1</td>
<td>A2</td>
<td>S1</td>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total # in the PC lab:</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>7</td>
<td>7</td>
<td>14</td>
<td>0</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CODE</th>
<th>EXPERIMENT</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Amplitude Modulation and Detection</td>
<td>Undergraduate Lab Ground Floor Printing House</td>
</tr>
<tr>
<td>C2</td>
<td>Frequency Modulation and Detection</td>
<td>Undergraduate Lab Ground Floor Printing House</td>
</tr>
<tr>
<td>C3</td>
<td>Digital Simulation Experiment 1</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>C4</td>
<td>Digital Simulation Experiment 2</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>D1</td>
<td>Analogue Simulation Experiment 1</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>D2</td>
<td>Analogue Simulation Experiment 2</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>A1</td>
<td>Signal Processing: Matlab and Basic Signal Analysis</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>A2</td>
<td>Digital Signal Processing: Fourier Analysis</td>
<td>PC Lab 1st Floor Printing House</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit Technology</td>
<td>Undergraduate Lab Ground Floor Printing House</td>
</tr>
<tr>
<td>OP</td>
<td>Optoelectronics Demonstration</td>
<td>Junior Sophister Lab, Physics Dept.</td>
</tr>
</tbody>
</table>

D students must do S1, S2, A1, A2, (if no space in Lab on Tuesday afternoon, then on Wednesday morning) in the first term/semester
CD students do S1 S2 A1 A2 T1 D1 D2. S1/2, A1/2 should be done in the first semester
C students do all. S1/2, A1/2 should be done in the first semester

Figure 2.6: JS Lab Groups and Timetable
2.4. COURSE DETAILS

2.4 Course Details

2.4.1 Signals and Systems (3C1)

Lecturers: Anil Kokaram and W. Coffey

Course Organisation

<table>
<thead>
<tr>
<th>Engineering Semester</th>
<th>Start Week</th>
<th>Hours of Associated Practical Sessions</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Total Lectures</th>
<th>Tutorials Per Week</th>
<th>Total Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

Total Contact Hours: 50

This course is taught in two parts. The first part [Introduction to Signal and System Analysis] occupies weeks 1-8 of the 1st Engineering Semester while the second [Introduction to Random Variables] occupies weeks 9-11 of that same semester.

Course Description  Signals and Systems is a one semester course taken by Junior Sophister C, CD and D Stream students. It provides a foundation for Signal Processing and Communications Engineering topics covered later in the undergraduate curriculum. The course is delivered in two parts. Introduction to Signal and System Analysis and an Introduction to Random Variables.

*Introduction to Signal and System Analysis* concentrates on building a foundation for understanding Signal Processing. In general Signal Processing is the study of the process of information extraction from signals such as images, audio, text or measurement data. The course introduces the student to methods for manipulating signals such as filters for hum removal. Both analogue and digital processing is considered. The student gains enough knowledge to be able to understand the basics of recent developments in media communications and visual media production.

*Introduction to Random Variables* concentrates on building the theoretical foundation for further work in Communications Engineering and Statistical Signal Processing. This part complements the material in the first part of the course. Most real signals are not deterministic e.g. sines and cosines, but are generated by random processes. Such processes take on values in time with some probability and hence can only be manipulated through analysis.
of the underlying probabilistic information generating process. This course introduces the student to the concept of Random Variables in manipulating stochastic signals. The classic problem of deciding whether a received pulse, corrupted by random noise, represents a 1 or a 0 is considered in full.

**Learning Outcomes**  On completion of this course the student will be able

- To analyse systems in order to calculate, estimate and classify their impulse, step, frequency response and evaluate their stability
- To analyse signals in order to calculate their frequency spectra, and estimate, classify, assess the effect of a system on signals in terms of frequency content and time domain effects
- To apply difference equations and the Z-Transform in calculating the output of a digital system given any digital input
- To analyse the effect of simple systems on random variables in terms of variance, correlation and probability
- To analyse, classify and compare Gaussian, Bernouilli, Binomial and Poisson random variables in terms of the underlying statistics
- To compute and estimate the probability of error in a communications system given the underlying signal statistics and decision system used

**Course Content**

- Systems Analysis [Dr. A. Kokaram]
  - Use of block diagrams, Differential Equation Models, What is a Linear Time invariant System?
  - Impulse response, convolution, step response, Laplace Transforms, transfer functions.
  - Poles, Zeros, Stability
  - Frequency response, Steady state response, Low pass and highpass filtering action
- Signal Analysis and Digital Signal Processing [Dr. A. Kokaram]
2.4. COURSE DETAILS

- Fourier series, Fourier Transform, Parseval’s Theorem
- Sampling theorem, Difference Equations and The z-Transform
- Low pass filtering, low pass filters, Basics of FIR, IIR Filters
- Stability and Applications of Digital Filters

• Introduction to Random Variables [Prof. W. Coffey]

- Expectation values, equivalence of time and ensemble averages for a stationary process, moments, autocorrelation function, spectral density, Wiener-Khinchin Theorem, Parseval’s Theorem, white noise, autocorrelation function and spectrum of filtered white noise.

- Probability distributions, binomial, Poisson and Gaussian distributions, Poisson and Gaussian distributions as limits of the binomial distribution, idea of an abstract probability space.

- Characteristic functions of random variables, central limit theorem, moments of statistical distributions from the properties of the characteristic function, properties of independent random variables, the multidimensional Gaussian distribution, Wiener process, random telegraph signal, Markov processes.

Teaching Strategies The strategy is a mixture of formal lectures, problem solving tutorials and interactive audio and video demonstrations. This is supported by two practicals (S1,S2) within the common Engineering Laboratories sessions. For the first part of the course, formal lectures are accompanied by notes containing gaps that student fill in as they attend the lectures. The notes are similar to workbooks in parts and students have to complete derivations and analysis during these lectures. During tutorials, students work on written analysis problems with the aid of the lecturer. The intention is to cover problems relating to the material covered in lectures during each week.

The lectures are punctuated by audio and video demonstrations that invite comment from the class about the effect that systems can have on signals. The intention is to connect in the students’ mind the mathematical analysis being undertaken with some intuition about the nature of signals. Two lectures are devoted to DSP applications in which basic image processing is demonstrated (edge detection), and the fundamentals of digital audio and
video compression are described. These lectures illustrate the importance of DSP in everyday consumer devices.

**Assessment** For CD and C students, 85% of the assessment is due to a three hour examination held during the Junior Sophister Engineering Part 1 Examinations at the end of the first semester. The remaining 15%, for CD and C students, are from the suite of Electrical and Electronic Engineering Laboratories held during the first and second terms. For D students the split is 90% and 10%, reflecting the reduced amount of EEE Laboratories that are undertaken in that stream specialisation.

**Recommended Texts**

- SIGNALS AND SYSTEMS, Oppenheim and Willsky, Prentice Hall,
- ELECTRONIC SIGNALS AND SYSTEMS, Paul A. Lyn, Macmillan Education
- MODERN CONTROL SYSTEMS, Dorf and Bishop, Addison Wesley
- SYSTEM ANALYSIS AND SIGNAL PROCESSING, Philip Denbigh, Addison-Wesley

**Further Information**

Web sites:
http://www.tcd.ie/Engineering/Courses/BAI/index.html

http://www.mee.tcd.ie/teaching/teaching.html#3c1Handouts

http://www.mee.tcd.ie/ sigmedia/teaching/3C1.php

ECTS Credits: 5.

**2.4.2 Electronic Engineering I (3C2)**

**Lecturer** Dr. M.J. Burke.
Course Organisation

<table>
<thead>
<tr>
<th>Semester or Term</th>
<th>Start Week</th>
<th>Hours of Associated Practical Sessions</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Lectures Total</th>
<th>Tutorials Per Week</th>
<th>Tutorials Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

Total Contact Hours: 50

Course Description, Aims and Contribution to Programme  *Electronic Engineering I* is a one semester course taken by Junior Sophister C, CD and D Stream students. It provides a thorough foundation in digital circuits as applied to modern logic device families. The course aims to provide junior sophister students with a knowledge of the operation and practical limitations of digital circuits at device, circuit and gate level. Students will learn the basic operational principles of the bipolar transistor and bipolar transistor based switching circuits as well as the performance limitations and design principles of logic circuits.

On completion of this course the student will be able:

- To explain the operation of the bipolar junction and MOS field effect transistors
- To analyse simple single-transistor switching circuits and to determine their performance criteria and limitations.
- To evaluate static and dynamic switching circuit parameters such as: logic voltage levels, drive currents, transition times, propagation delay and power dissipation.
- To explain the internal operation and identify the logic function performed by a 7400 series TTL gate from its circuit structure.
- To determine the voltages at all nodes within a TTL gate circuit and to evaluate its power consumption.
- To design simple synchronous finite state machines.

Content of Course  Semiconductor Materials: revision of fundamental laws; carrier transport phenomena; current flow mechanisms; the p-n junction; barrier potential; the ideal diode equation.
Bipolar Junction Transistor: physical principles of operation; device characteristics and parameters.
Bipolar Transistor Inverter: operation of the transistor as switch; simple inverter circuit; static and dynamic performance characteristics; effects of loading.
TTL Logic Family: logic characteristics and performance; operating principles of standard 7400 series gates; circuit analysis and power consumption evaluation.
MOS Field Effect Transistor: physical principles of operation; device characteristics and parameters.
Synchronous Finite State Machines: modelling and design of synchronous finite state machines.

Teaching Strategies The teaching approach is primarily a combination of lectures and problem solving tutorials. This is supported by two practical laboratories (D1 and D2, though the analogue experiments A1 and A2 are equally relevant) within the common Engineering Laboratories sessions.

The students are issued with a comprehensive set of course notes which contain strategic gaps that are filled in during lectures. The notes contain diagrams identical to those used on overhead view-foils during lectures and textual explanation which is identical to that given orally. This allows the student to spend much time during lectures listening and being able to assimilate information without having to transcribe everything that is written on the blackboard or spoken by the lecturer giving the opportunity for in-class learning.

Many of the mathematical derivations are missing from the notes and these are done on the blackboard during class and filled into the gaps in the notes. This allows analytical work to be done interactively with participation from the class. It also keeps the students in active mode and occupied to maintain interest and class quietness. The notes are of little value without the crucial derivations so this also encourages lecture attendance.

During tutorials, students work on analytical and design problems with the aid of the lecturer. Each week a sheet of 3 or 4 problems is issued which relates closely to the material covered in lectures a week earlier. The lecturer observes the progress of each student as he/she works through the problems, giving advice and assistance as required. Global assistance may be given to the whole class interactively using the blackboard as required. Students are
left to complete the problems in their own time in conjunction with revision of their lecture notes. Fully worked solutions to each problem sheet are then made available on a departmental server approximately two weeks after each sheet is issued so that students can compare these with their own attempts. Students are given the option of submitting their attempts at the problems for correction by the lecturer but this is rarely taken up and in-class progress is usually satisfactory.

**Assessment**  For CD and C students, 85% of the assessment is based on a three hour examination held during the Junior Sophister Engineering Part 1 Examinations at the end of the first semester. The remaining 15%, for CD and C Stream students, is based on the suite of Electrical and Electronic Engineering Laboratories held during the first and second terms. For D Stream, students the split is 90% based on the examination and 10% on laboratories, reflecting the reduced amount of EEE Laboratories that are undertaken by the latter.

**Recommended Texts**


**Further Information**
ECTS Credits: 5.

**2.4.3 Electronic Engineering II (3C3)**

**Lecturer**  Professor J.K. Vij/ Professor W. T. Coffey.
Course Organisation

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Lectures Total</th>
<th>Tutorials Per Week</th>
<th>Tutorials Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13</td>
<td>24</td>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

Total Contact Hours: 50

Course Description, Aims and Contribution to Programme  Electronic Engineering is a one semester course taken by Junior Sophister C Stream students. It provides a foundation for Analog Electronic Engineering for the topics covered later in the undergraduate curriculum. When the students come to this course, they have only limited knowledge of the active components of an electronic circuit. The student is introduced to bipolar and FETs and their equivalent circuits. The basis parameters of devices are introduced. This introduction leads to their use in basic amplifiers of various specifications. The students are then introduced to the operational amplifiers and their limitations. Various circuits from basic circuits of addition, subtraction, amplification to those involving specific applications are dealt with. The circuits involving current to voltage and voltage to current converters are given.

This leads to the course on the design of oscillators. Several types of oscillators are studied. Analog computers both in the block diagram as well those involving hardware are studied with a view to advancing the use of integrated circuits and systems to analogue circuits. The last part of the course involves the various active filters. For these, the students need to have a good grasp of the electrical circuit theory and they are brought to the stage where they can actually design the filters with the various specifications and can write and solve equations for the circuit.

Learning Outcomes  On completion of this course the student will be able

- To design amplifiers and oscillators using discrete devices and integrated circuits
- To use basic negative and positive feedback theory to explain stability and instability in electronic circuits
- The advanced electrical circuit theory: time and frequency domains
- To design A/D and D/A converters
2.4. COURSE DETAILS

- To design active filters
- To design analog computers for solving a second, third and fourth order differential equation.
- To design analog active filters with a 2nd order transfer function for low pass, band pass and high pass filters

Content of Course  There are seven main themes covered in the course as follows.

- Small Signal analysis and Design of amplifiers, differential amplifier using discrete components
- Design of RF amplifiers
- Oscillators
- Operational Amplifiers and its applications in amplification, oscillation, addition, subtraction, differentiation, integration, analog computing, multiplier and differential amplifier
- Analogue-Digital Converters
- Digital to Analogue Converters
- Active Filters

Teaching Strategies  The strategy is a mixture of lectures, problem solving tutorials. This is supported by one mini project on analog electronics (3C6). For the first part of the course, formal lectures are accompanied by notes given only for difficult parts. During tutorials, students work on written analytical problems with the aid of the lecturer. The intention is to cover problems relating to the material covered in lectures during each week. The lecturer supervises each student as they work through problems, giving advice and assistance as required. During the tutorial, the lecturer illustrates the essence of the solution to each problem once the class has made a realistic attempt.
Assessment  For CD and C students, 85% of the assessment is due to a three hour examination held during the Junior Sophister Engineering Part 1 Examinations at the end of the first semester. The remaining 15% are from the suite of Electrical and Electronic Engineering Laboratories held during the first and second terms.

Recommended Texts

- Integrated Electronics, Jacob Millman and Christos C. Hakias, McGraw Hill,


- Analog to Digital and Digital to Analog Converters, Analog devices

Further Information

Web site:  
http://www.tcd.ie/Engineering/Courses/BAI/.  
ECTS Credits: 5.

2.4.4  Electromagnetism and Optoelectronics (3c4)

Lecturers:  Professor W. T. Coffey, Dr. Louise Bradly

Duration:  Three 1-hour lectures and one 1-hour tutorial per week, Semester 2

Electromagnetism

Electrostatics, Coulomb.s Law, Laplace.s equation, Poisson.s equation, capacitance, magnetostatics, Ampere.s theorem, magnetic vector potential, displacement current, plane-waves, Maxwell.s equations, boundary conditions, reflection and refraction at a boundary, simple treatment of radiation, transmission lines.
Optoelectronics

The quantum physics section of the course consists of three parts: Introduction to Wave Mechanics, Wave Mechanics of Electrons in Crystals and Devices. It begins with some fundamental, empirical aspects of quantum physics such as the de Broglie hypothesis and blackbody radiation. Then the idea of energy quantisation in matter is introduced through topics such as solution to the Schrodinger equation for a particle in a 1-D box.

The second part of the quantum physics course deals with electrons in periodic potentials and in the last part, material from the first two parts is used to discuss some simple devices.

Courses 3C4 and 4C11 are the basis for understanding the technology of optoelectronics and its application to high speed information technology, specifically, optical fibre communications and optics in computers. The courses take the student from a basic knowledge of optics all the way to systems considerations. Application of Maxwell’s equations to derive Fresnel’s equations of reflection at single and double interfaces. Examples of devices such as Fabry-Perot cavities, total internal reflection guiding, polarisation control. Diffraction theory and its application to limits of lithography and imaging.

Assessment: A three hour examination held during the Annual examination period.

(Reference) N. Gershenfeld The Physics of Information Technology, Cambridge University Press London 2000 (chapter 5)

2.4.5 Telecommunications (3C5)

Lecturer F. Boland and P.C. Fannin

Course Organisation

<table>
<thead>
<tr>
<th>Engineering</th>
<th>Start</th>
<th>Hours of Associated</th>
<th>End</th>
<th>Lectures</th>
<th>Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester or Term</td>
<td>Week</td>
<td>Practical Sessions</td>
<td>Week</td>
<td>Per Week</td>
<td>Total</td>
</tr>
<tr>
<td>Semester 2</td>
<td>13</td>
<td>6</td>
<td>24</td>
<td>3</td>
<td>33</td>
</tr>
</tbody>
</table>

Total Contact Hours: 50
CHAPTER 2. JUNIOR SOPHISTER YEAR

This course is taught equally by two lecturers over a period of 11 weeks with each lecturer.

Course Description, Aims and Contribution to Programme  Telecommunications is a one semester course taken by Junior Sophister C and CD Stream students. Its objective is to give the students a basic understanding of telecommunications including an introduction to digital communications and information theory. The course consists of two components, introduction to Telecommunications and introduction to Digital Communications.

Introduction to Telecommunications concentrates on building a foundation for an understanding of Telecommunications. The course introduces the student to the concepts of, i) modulation and demodulation of amplitude and angle modulated waves and ii) time and frequency division modulation methods. The concepts of baseband, bandwidth, bandlimiting, narrowband and wideband signals are also discussed. Digital modulation is introduced and a single channel PCM system is thoroughly covered with the relevance of sampling and aliasing being highlighted.

Introduction to Digital Communications addresses both the introductory principles of digital information coding and transmission and the foundations of information theory. Building on the students knowledge of sampled data systems, as introduced in 3C1 Signal & Systems, the course examines the quantization of signals and considers the influence of wordlength, amplitude distribution and acquisition time on coding accuracy. The course builds on the knowledge of probability theory and random processes also introduced in 3C1. Then the course introduces basic information theory and coding examining topics including entropy measures of information, channel capacity and source coding.

Learning Outcomes  On completion of this course the student will be able

- To classify AM and FM signals and methods of generation and detection of such signals.
- To analyse modulated signals in order to calculate their frequency spectra and resulting bandwidth. To understand and appreciate the associated terms, baseband and bandlimiting.
- To demonstrate an understanding of digital modulation and be able to contrast such a system with analogue systems.
2.4. **COURSE DETAILS**

- To quantify the effect of system parameters on the digitization of signal.
- To calculate the information content of a signal from its probability distribution.
- To define channel capacities and the construction of efficient codes for communication.

**Content of Course**

- AM systems and techniques
- FM systems and techniques
- Basic introduction to digital modulation
- Introduction to source encoding
- Introduction to information theory
- Basic coding theory

**Prof. P.C. Fannin.**


- Pulse amplitude modulation, Pulse code modulation, Sampling, Aliasing.

- Multiplexing: Frequency division and Time division multiplexing.
Prof F. Boland

- Signal quantization and quantization errors.
- Mathematical foundations of information theory
- Entropy as the measure of information
- Source coding, error correction and channel capacity

Teaching Strategies
The strategy is a mixture of lectures and problem solving tutorials. This is supported by two practicals (T1,T2) within the common Engineering Laboratories sessions. During tutorials, students work on written analysis problems with the aid of the lecturer. The intention is to cover problems relating to the material covered in lectures given during the previous week. During the tutorial the lecturer leads the students through the problems, giving advice and highlighting potential difficulties of the problems. Two laboratories from the suite of Electronic Laboratories are particularly relevant to supporting the work in 3C5. These are T1 and T2. They cover the basics of Amplitude modulation and Frequency modulation.

Assessment
For CD and C students, 85% of the assessment is due to a three hour examination held during the Junior Sophister Engineering Part 1 Examinations at the end of the first semester. The remaining 15%, for CD and C students, are from the suite of Electrical and Electronic Engineering Laboratories held during the first and second terms. For D students the split is 90% and 10%, reflecting the reduced amount of EEE Laboratories that are undertaken in that stream specialisation.

Recommended Texts
- *Electronic Communication Techniques*, P.M. Young.
- *Introduction to Communications Systems*, F.G. Stremler.
Further Information

http://www.tcd.ie/Engineering/Courses/BAI/
ECTS Credits: 5.

2.4.6 Electronic Systems Design & Implementation Project (3C6)

Lecturers  Professor J. K. Vij (Analogue Project), Dr. A. Moore & Mr. R. Dempsey (Digital Project)

Course Organisation  This course is comprised of two parts, one in the first term—Michaelmas term—and one in the second term—Hilary term. In the Michaelmas term, the C-stream Junior Sophister students undertake an electronics project in analogue circuitry, coded as 3C6A. In the Hilary term, both C-stream and CD-stream students are given a digital electronics project, coded 3C6B.

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>Hours of Associated Practical Sessions</th>
<th>End Week</th>
<th>Practicals</th>
<th>Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term 1</td>
<td>2</td>
<td>–</td>
<td>9</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>–</td>
<td>17</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
<td>Total</td>
</tr>
<tr>
<td>Total Contact Hours: 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Course Description  The main purpose of the two projects is to develop the students’ practical knowledge of the design, simulation, implementation, and testing of analogue and digital circuits. Students learn to work in a group of usually four persons, who must manage the project, divide up the workload and apply a ‘divide-and-conquer’ approach to tackling a design task in Electronic Engineering—analogue in the first term and digital in the second term. In each case, the circuit should be assembled in modular form and each section tested before proceeding further. This simplifies the task of verifying and modifying the design. Students are given a worksheet detailing a ‘Schedule of Activity’ and each week the group uses a new work-schedule sheet wherein members sign off their names against a particular project activity so that progress or holdups can be tracked.
CHAPTER 2. JUNIOR SOPHISTER YEAR

Learning Outcomes

**Analogue Design Project (3C6A)**  On completion of this course the student will be able to:

- Describe and plan a project involving analog electronics
- Write a specification of the project
- Sketch a block diagram of the overall circuit along with inputs, and outputs
- Select a definite test strategy to check each stage of the design
- Describe operation, collect data on all aspects of the project, and write a detailed technical report.
- Construct a hardware solution for an analog electronics problem.

**Digital Design Project (3C6B)**  On completion of this course the student will be able to:

- Describe and plan a project involving digital electronics
- Write a specification of the project
- Sketch a block diagram of the overall circuit along with inputs, outputs and user interfaces
- Select a definite test strategy to check each stage of the design
- Describe timing waveforms, collect notes on all aspects of the project,
- and finally, write a detailed technical report.
- Construct a hardware solution for a digital electronics problem.
Digital Design Project (3C6B)

- Digital Electronics
  - Fundamental building blocks of digital circuits from CMOS gates to system level devices.
  - Frequently used important functions like decoders, multiplexors, flip-flops, registers and shift registers, counters and timers.
  - Use of block diagrams, circuit schematics with PSpice, circuit simulation and testing.
  - Using programmable logic to simplify designs, reduce cost, and handle rapid modifications.

- Analysis and design of combinational and synchronous digital systems.
  - Hardware construction of two real working circuits required., one introductory experiment and one more complex system and both at the chip level.
  - Investigating signals and timing to verify circuit operation.

- Engineering documentation
  - Good documentation is essential for correct design and efficient maintenance of digital systems. While undertaking the project in teams the individual students must keep notes on the progress of their work on the associated project tasks. The following list gives the crucial items to be documented and to feature prominently for explanation in the students report.
    - A circuit specification.
    - A block diagram with basic interconnections
    - A schematic diagram of the electrical components of the system
    - A timing diagram illustrating cause and effects between signals
    - A thorough description of the operation of the circuit.

Teaching Strategies
Analogue Design Project (3C6A)  Students learn progressively more difficult design methods and tools as the weeks progress. Thus in the first weeks, the students are heavily assisted to design, simulate, construct and test a simple introductory analog circuit based on an Operational Amplifier. The essential steps in the design procedure are described and illustrated. They see the important role of calculations in circuit design and the problems associated with testing the design by making use of professional tools. The students are assigned a mini-project. At the present time, there exist five projects as follows:

- **Audio Amplifier Design**: The objectives here are (i) to design, build and test an amplifier to reproduce a microphone input through a speaker, and to (ii) design, build and test a simple tone control for the amplifier.

- **An Audio Alarm**: The objective is to design, build and test an alarm circuit using closed loop switches. The output should be generated by outputting a suitable waveform, via a power amplifier, through a speaker.

- **A Raster Display**: The objective is to design a raster display of 9 horizontal lines on the screen of an oscilloscope and to remove any fly-back lines by using Z blanking.

- **A Sine Wave Oscillator**: The objective is to design this system using a Wien bridge Oscillator and associated circuitry for stabilizing the amplitude of oscillations and arrangement for varying the frequency of the generated sine wave.

- **A Function Generator**: The objective is to design a system for creating square, sine and triangular waves.

The students are allocated one of the projects given above in the first week of the term. They are asked to note the relevant specifications, instructions and consult the literature and then come up with the design of the circuit and systems themselves. Before proceeding to build the circuitry, they need to consult the instructor. After the design is cleared by the instructor, the group of students assemble the circuitry on a breadboard, and test it. If the circuit works, they are encouraged to produce a working prototype of the system by soldering the circuit elements and the devices on a breadboard. The students have an access to the general test equipment such as power supplies,
oscilloscopes, multimeters etc. and the mechanical tools in the laboratory. Individual help is made available as the instructor regularly enquires about the progress of the work. Apart from the first one week, the students are timetabled to work in the laboratory three hours per week.

**Digital Design Project (3C6B)** As with the Analogue Design Project, in the Digital Design Project, students learn progressively more difficult design methods and tools as the weeks progress. The students are heavily assisted in the initial weeks to design, simulate, construct and test a simple introductory digital circuit based on a CMOS counter IC. The essential steps in the design procedure are described and illustrated. They see the important role of simulation in circuit design and the problems associated with testing the design by making use of professional tools, like Pspice. Next, the students are presented with the problem of designing a far more complex system. In this case we have decided to build a decoder and display for a digital clock based on the Rugby (MSF) time code. The students have an option here of choosing between a hardware solution using CMOS chips or an implementation using a PIC processor system. The latter is favoured by those students more inclined to software solutions.

Teaching is conducted in the CadLab of the EEE Department. Students sit at PCs fitted with the required hardware and software. The instructor uses the teaching PC to demonstrate the logistics of using Pspice and also demonstrate design issues to the entire class. Individual help is then made available as the instructor moves around the groups giving advice and direction as needed. Full laboratory facilities are available including power supplies, signal generators, oscilloscopes, breadboards, tools and a component stock with data sheets. All these provide a hands-on experience for the digital designer.

**Assessment** Each component of 3C6 is marked out of 50. The Analogue Design Project—3C6A—is marked as follows:
### Analogue Design Project (3C6A) Marking Scheme

The Digital Design Project—3C6B—is marked similarly, as follows:

<table>
<thead>
<tr>
<th>Category</th>
<th>Items &amp; Marks</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Presentation</td>
<td>General Layout &amp; Plan</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Problem Overview &amp; Spec</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Conclusions &amp; Extras</td>
<td>2</td>
</tr>
<tr>
<td>Report Contents</td>
<td>Contents &amp; Plan</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Block Diagrams &amp; Spec</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Circuit Design &amp; Schematics</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Circuit Operation</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Test Strategy</td>
<td>8</td>
</tr>
<tr>
<td>Circuit Implementation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Digital Design Project (3C6B) Marking Scheme

#### 2.4.7 Microprocessor Systems 1 (3D1)

**Lecturer**  Mike Brady.

**Course Organisation**  The course runs for all eleven weeks of Engineering Semester 1 and comprises three lectures and one tutorial per week, with a two-hour practical in each of the last seven weeks. Total contact times is thirty three hours lectures, eleven hours tutorials and fourteen hours practicals, fifty eight hours total.
2.4. COURSE DETAILS

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Lectures Total</th>
<th>Tutorials Per Week</th>
<th>Tutorials Total</th>
<th>Practicals Per Week</th>
<th>Practicals Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>14</td>
</tr>
</tbody>
</table>

Total Contact Hours: 58

Course Description, Aims and Contribution to Programme  

*Microprocessor Systems 1* is a one-semester course taken by Junior Sophister C-, CD- and D-Stream students. It covers the Instruction Set Architecture (ISA) of a typical microprocessor-based computer—the Motorola MC68000, and equips the student with a knowledge of the architecture, the associated assembly language, input/output programming techniques, exceptions (including interrupts) and exception-handling techniques. The module concludes with a simple introduction to the bus and instruction timing estimation.

The course is intended to enable students to design and develop programs and program ‘architectures’, to test and debug programs and to analyse and modify their execution behaviour, based on a thorough familiarity with the low-level architecture of a computer. Concepts such as RISC/CISC architectures, register sets, addressing modes, data structures, subroutines, [informal] high-level to low-level language translation techniques, polling, interrupt priorities, asynchronous producer-consumer systems are introduced.

Learning Outcomes

- The student will be able to analyse, specify, design, write and test assembly language programs of moderate complexity.

- The student will be able to select an appropriate ‘architecture’ or program design to apply to a particular situation; e.g. an interrupt-driven I/O handler for a responsive real-time machine. Following on from this, the student will be able to design and build the necessary programs.

- The student will be able to calculate the worst-case execution time of programs or parts of programs, and to design and build, or to modify, software to maximise its run time memory or execution-time behaviour.

- The student will be able to characterise and predict the effects of the properties of the bus on the overall performance of a system.
• The student will be able to describe some of the characteristics of RISC and CISC architectures.

Content of Course

• Review of Binary and Hexadecimal Arithmetic
• The Von Neumann Machine
• The Programmer’s Model of the MC68000
• Data Representation: integers, characters, signed representations, arrays
• Addressing modes: immediate, direct, indirect
• Program flow control: unconditional branch and jump
• The Condition Code Register
• Conditions and conditional branching
• High-level language constructs: while, if, for, etc.
• Some complex 68000 instructions
• Subroutines: mechanisms and parameter passing
• Principles of Input/Output: polling
• Exceptions and exception handling
• Supervisor & user mode
• Interrupts and interrupt handlers
• Producer consumer organisation; queues and buffers
• Introduction to the System Bus
• Instruction Execution
• Instruction Timing
2.4. COURSE DETAILS

Teaching Strategies  The teaching strategy is a mixture of lectures, problem-solving tutorials and hands-on practicals. The format of lectures is conventional; however, a great deal of informal interaction is normal, and students can expect to participate in question-and-answer and problem solving sessions. For the first four weeks or so, the students are taught the general principles of low-level architecture and programming. Tutorials held during this time review basic skills such as binary and hexadecimal, algorithm design and challenge the students to build programs based on a partial knowledge of the computer’s instruction set. Practicals, starting in the fourth week, require the students to design, write, evaluate and debug their programs on special-purpose development systems. More advanced topics introduced during lectures become the subject of practicals through the rest of the semester.

Assessment  Assessment is by examination and by practical. Practicals attract a mark of up to 20% of the year end mark, and the examination make up the remaining 80% or more.

The practicals, conducted by each student individually, encourage the design, writing and testing of programs and the development of the skills needed in actual practice. Practicals are run by postgraduate students—demonstrators—who are expert in the subject. The practicals are highly formative, with students receiving feedback and advice from the demonstrators, who also mark the student’s work at the end of the session. The mark makes a small contribution to the student’s end-of-course result, amounting, in total, to not more than 20% of the year-end mark. The practicals particularly help with the first two learning outcomes, and make a significant background contribution to the third.

The examination is three hours long, and students are required to answer five questions from a selection of seven. Most questions will contain a short discursive component and a related question requiring the student to demonstrate an ability to design and write code. Some questions might ask the students about bus architecture or the effects of bus performance, requiring worked examples to be furnished.

Recommended Texts  Any introduction to 68000 programming, such as by King & Knight, Bacon, Clements, etc.
CHAPTER 2. JUNIOR SOPHISTER YEAR

Further Information
Web site: 
http://www.tcd.ie/Engineering/Courses/BAI/JS_Subjects/3D1/
ECTS Credits: 5.

2.4.8 Microprocessor Systems 2 (3D2)

Lecturer  Dr Jeremy Jones.

Course Organisation  This course runs for the 11 weeks of Engineering Semester 2. It comprises 3 lectures and 1 tutorial per week. In addition, for the final 6 weeks (Trinity Term) there is a 3 hour practical on Friday afternoons.

<table>
<thead>
<tr>
<th></th>
<th>Lectures</th>
<th>Tutorials</th>
<th>Practicals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
<td>Total</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Course Description, Aims and Contribution to Programme  This course, which naturally follows 3D1, is divided into two parts (1) the design, construction and debugging of a simple microprocessor system and (2) the architecture of pipelined RISC CPUs.

In the first part of 3D2, students receive a theoretical introduction to the fundamental elements of a modern microprocessor system which they then have to put into practise by constructing a working microprocessor system comprising 68008 CPU, EPROM, RAM, ACIAs, glue logic and a basic monitor program. The course aims are (i) to give students enough knowledge so that they can design, construct and get their own microprocessor system working (ii) to make students realise that such systems are deterministic and do not work my “magic”, (iii) that an attention to detail is important when dealing with hardware and software at this level, (iv) that logically based strategies need to be applied to locate faults whether they are at a design or constructional level and (v) to have exposure to working as a member of a group.

In the second part of 3D2, students receive an introduction to the architecture of RISC CPUs, instruction level pipelining and how data, load and control hazards can be resolved effectively. Students make use of an interactive web based animation of a DLX/MIPS CPU to explore these topics.
2.4. COURSE DETAILS

Learning Outcomes  Students will be able to

- Design simple logic circuits using programmable logic.
- Design, construct and debug a simple microprocessor system.
- Apply logically based strategies for locating faults.
- Have experience working together as a group, keeping accurate documentation and writing an engineering report.
- Explain the RISC design philosophy and translate high level language programs onto a RISC instruction set.
- Explain the key concepts behind instruction level pipelining and know how to apply a number of techniques to overcome data, load and control hazards.

Content of Course

Part 1

- Review of totem-pole, tri-state & open-collector outputs.
- Logic design using programmable logic.
- Hardware review of the 68008 microprocessor.
- Interfacing the 68008 with memory and 6800 peripherals devices.
- Monitor implementation.
- Interrupt driven I/O.
- Design, construction and debugging of a simple 68008 based microprocessor system.
Part 2

- RISC vs CISC.
- RISC-1 design criteria and architecture.
- Register windows and delayed jumps.
- Instruction level pipelining.
- The DLX/MIPS pipeline.
- Resolving data, load and control hazards.

**Teaching Strategies**  The teaching strategy is a mixture of lectures, problem solving tutorials and six 3 hour practicals. A key component is the “hands-on” group project to construct a simple microprocessor system from lecture notes and a kit of parts using wire-wrapping. Students have monitored weekly targets which they should try and meet. Students can avail of help from the course lecturer, teaching assistants and from each other during the 3 hour weekly practical. Students learn how use oscilloscopes and logic analysers to collect evidence in order to formulate logical strategies for fault location and correction. This is the most interesting and challenging part of the course.

Students make use of an interactive web based animation of a DLX/MIPS CPU to explore the DLX/MIPS pipeline and how data, load and control hazards are resolved.

**Assessment**  The group project account for 20% of the final mark and the exam 80%. Students must answer 5 out of 6 exam questions (four from part 1 and two from part 2).

The group project is marked from their weekly progress reports and a group project report. All members of the same group normally receive identical marks.

**Recommended Texts**

- *Microprocessor Systems Design*, Alan Clements
2.4. COURSE DETAILS

- *Computer Architecture—a Quantative Approach*, John Hennessey & David Patterson

- *High Performance Computer Architecture*, Harold Stone

Further Information
Web site:
ECTS Credits: 5.

2.4.9 Concurrent Systems (3D3)

Lecturer .

Course Organisation The course runs for all eleven weeks of Engineering Semester 2 and comprises three lectures and one tutorial per week, with a one-hour practical in eight weeks. Total contact time is thirty three hours lectures, eleven hours tutorials and eight hours practicals, fifty two hours total.

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Lectures Total</th>
<th>Tutorials Per Week</th>
<th>Tutorials Total</th>
<th>Practicals Per Week</th>
<th>Practicals Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>33</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

Total Contact Hours: 52

Course Description, Aims and Contribution to Programme *Concurrent Systems* is a one-semester course taken by Junior Sophister CD- and D-Stream students. It is divided into two parts, concurrent programming and an introduction to operating systems.

A concurrent system consists of multiple smaller programs that allow it to perform multiple computations simultaneously and to control multiple external activities which occur at the same time. Sharing data in a concurrent environment is fraught with peril, errors only manifest themselves sporadically and are not easily reproduced. This section of the course is based around the concepts, models and practice of writing safe concurrent computer programs. The concepts of how to safely share data and deal with the associated problems of deadlock, livelock and starvation are studied in detail.
Throughout the course concurrent programs are modelled as Finite State Process models. Concurrent programming errors can be effectively specified, reproduced and analysed in this framework. Students learn how to construct Finite State Process models and how to test these models for safety and liveness violations. The course then provides students with the skills to analyse any concurrent programming errors present, and to construct a new safe model. This correct model is then put into practice as a system of Java threads. The emphasis is on producing a correct design of the system through a modelling and rigorous checking process before implementation.

The second part of the course focuses on the parts of operating systems that allow and support the writing of concurrent programs. An introduction to operating systems is provided and processes, threads and scheduling policies are considered in more detail. The emphasis of this part of the course is the different ways in which operating systems can affect the execution of concurrent programs.

Learning Outcomes

• The student will be able to construct a Finite State Process model from a problem description, specification or implementation of a concurrent system.

• The student will be able to specify and analyse any safety or liveness violations in the model.

• Based on this analysis the student will be able to construct a correct model for the system.

• The student will be able to use this correct model to design and write an implementation of the system as a Java program.

• The student will be able to describe, interpret and predict the effects of the operating system on the execution of a concurrent program.

Content of Course

• Concurrent Execution

• Concurrent Systems modelled as Finite State Processes
2.4. COURSE DETAILS

- Implementation of Finite State Processes as Java Threads
- Shared Data
- Mutual Exclusion
- Semaphores
- Monitors
- Condition Synchronization
- Deadlock
- Safety and Liveness Properties
- Model Based Design
- Operating Systems Overview
- Processes
- Threads
- Scheduling

**Teaching Strategies**  The teaching strategy is a mixture of lectures, problem-solving tutorials and practical sessions. The notes accompanying the course contain gaps which the students fill in during lectures. Questions are welcomed during lectures and discussion is encouraged when appropriate topics are being covered.

Students work on written problems during tutorials. The problems reinforce material presented during lectures and involve specifying Finite State Process models and writing and analysing Java code. Help and guidance is provided during the tutorials and a combination of white board and computer are used to work through and demonstrate the solutions.

The practicals, conducted by each student individually, encourage the design, modelling, and writing of concurrent programs and the development of the skills necessary to produce correct concurrent programs. The practical sessions are very hands-on and students are expected to produce working Finite State Process models and Java programs. Students receive feedback and direction each week during the practical sessions.
Assessment  Assessment is by examination and by practical. Practicals carry a mark of 20% of the year end mark, and the examination make up the remaining 80%.

Three of the more difficult practicals are completed over several weeks and account in total for 20% of the overall mark.

The examination is three hours long, and is divided into two sections. Section A contains 32 multiple choice answer questions and is worth 40% of the total mark. Section B contains 4 questions and is worth 60% of the examination mark. Students are required to answer all the questions in section A and three questions from section B. The multiple choice questions cover the whole course and require students to demonstrate knowledge, the ability to identify concurrent programming errors and the ability to analyse Finite State Process models. Section B requires the student to construct Finite State Process models and design and write Java implementations of concurrent systems. Some questions may also ask about operating system concepts and policies.

Recommended Texts

- *Concurrency: State Models Java Programs*, Magee and Kramer, John Wiley & Sons


Further Information
Web site:  
http://www.tcd.ie/Engineering/Courses/BAI/JS_Subjects/3D3/.  
ECTS Credits: 5.

2.4.10  Computer Aided Design 1 (3D4)

Lecturer  Carol O’Sullivan and John Dingliana.

Course Organisation  The course runs for eleven weeks of Engineering Semester 2 and comprises three lectures and one practical per week. Contact time is thirty three hours lectures and eleven hours practicals; forty four hours total.
## Course Description, Aims and Contribution to Programme

*Computer Aided Design 1* is a one-semester option which may be taken by Junior Sophister CD- and D-Stream students.

The objective of this course is to equip the students with the fundamental knowledge of the major elements of CAD and Computer Graphics and to help them explore related areas including design, geometric modelling, rendering and animation. Students are introduced to the standard architectures of modern graphical applications including details on the underlying hardware and low-level software components of such systems. The course is intended to enable students to bridge the gap between these low-level fundamental, components common to all computer applications, and the high-level abstract output in most interactive graphical applications.

Students are also introduced to OpenGL, a modern high-level graphics API which is widely used for 3D Design and Visualisation, and this is used throughout the course to demonstrate concepts and to allow the students to develop their own 3D applications.

Finally, some advanced topics are introduced. Firstly, advanced modelling and representation techniques are presented and parametric techniques for modelling curves and surfaces are covered in detail. Global illumination techniques are discussed and the raytracing method is presented in detail. Finally, animation principle and methods are taught.

## Learning Outcomes

- Students will be able to identify, define, recognize and state the underlying details common to graphical applications including primitive data structures, low-level algorithms, I/O processes and hardware.

- Students will be able to classify, discuss, recognize, explain and use the different stages of the 3D graphics pipeline including, Modelling, Transformation, Viewing, Projection, Hidden Surface removal, local illumination and shading algorithms.
• Students will be able to use and apply the OpenGL API and organise standard Graphical Data Structures in order to develop basic 3D graphical applications and produce useful and creative graphical output.

• Students will be able to model complex objects using parametric definitions and render these realistically using the OpenGL API. They will also be able to derive the mathematical formulations for these surface representations.

• Students will be able to construct a scene and camera system for the purposes of global illumination, describe the illumination model used to render this scene with the ray tracing algorithm, outline the algorithm in detail, construct eye rays and find their intersections with implicitly modelled scene objects.

• Students will be able to list and describe the classes and principles of animation. They will be able to define, describe, contrast and compare Forward and Inverse Kinematic techniques.

Content of Course

• An Introduction to CAD and Computer Graphics. Problem Domain and Applications.

• Computer Graphics Hardware. Input and Output Devices.

• Colour in Computer Graphics

• An introduction to the OpenGL API for 3D computer graphics

• The Computer Graphics Pipeline

• Three Dimensional Transforms

• Projection and Viewing

• Hierarchical 3D Transformations

• Window to Viewport Transformation

• Basic Rasterization: Line Drawing and Anti-aliasing
2.4. COURSE DETAILS

- Hidden Surface Removal Techniques
- The Phong Illumination Model
- Shading Algorithms
- Parametric Cubic Curves and Bicubic Surfaces
- Ray Tracing
- Animation: Classification and Principles
- Forward and Inverse Kinematics

Teaching Strategies  The teaching strategy is a combination of lectures and hands-on practicals. The format of lectures is conventional, with a great deal of practical examples to emphasize the concepts that are discussed in course note and recommend texts. There is a small degree of informal interaction in the lectures but students have an opportunity for more direct interaction in the practical sessions throughout the course.

The first half of the course introduces students to the fundamental principles of CAD and Computer Graphics. Practicals during the first four weeks deal directly with issues discussed in lectures and give the students an opportunity to use their newly learnt knowledge in developing their own small applications. During this early period, a significant amount of guidance is provided, by the lab demonstrators and from the practical notes, in getting acquainted with the new OpenGL API. However students are required to produce some degree of individual creative output in each of the practical exercises.

The second half of the course introduces some more advanced topics in Computer Graphics, broadly categorized into the areas of Modelling, Rendering and Animation. The students complete one larger project during this part of the course, that integrates the knowledge gained to date. They receive support and advice in this task during the weekly laboratory sessions.

Assessment  Assessment is by examination and practical coursework. End of year examinations make up 80% of the final mark, whilst the total coursework contributes to the remaining 20%.

The practical exercises are conducted individually by each student every week, in the presence of a postgraduate demonstrator, who is expert in the
subject and marks the students work. In most cases the practical task is set a week before the student has to demonstrate it in labs, although in some cases larger practical tasks can be set for a two or more weeks development period. For the second part of the course, a larger project is assigned, on which they work for several weeks.

**Recommended Texts**

- *Introduction to Computer Graphics* Foley, Van Dam, Feiner, Hughes, Phillips


**Further Information**

ECTS Credits: 5.

**2.4.11 Software Design and Implementation (3D5)**

**Lecturer** Glenn Strong.

**Course Organisation** The course runs in two parts. Part A, taken by D- and CD-stream students runs for all eleven weeks of Engineering Semester 1 and comprises one tutorial per week, and one three-hour practical in each week. Total contact times is forty-four hours total.

The second part is taken by D-stream students only and runs for the second semester on the same schedule as part 1, for a total of fourty-four contact hours.

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures</th>
<th>Tutorials</th>
<th>Practicals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Total Contact Hours: 88
2.4. **COURSE DETAILS**

**Course Description, Aims and Contribution to Programme** *Software Design and Implementation* is a two-semester course; the first semester is taken by CD-Stream and D-Stream students, and the second semester by D-Stream students only (CD-Stream students take 3C6 in the second semester). The course covers Software Engineering practice through one or two semester-long programming projects. Students will gain practical experience with good software engineering techniques through the design, implementation and testing of a software application.

The course is intended to give students direct experience with programming in the large, to teach application of techniques for program design, software implementation, testing and documentation. Concepts such as requirements capture, the software engineering life cycle, object-oriented software design, the design of the Java programming language, GUI implementation in Java, project management, software testing, selection of appropriate algorithms and data structures for problem solving, and basic concurrency and networking are introduced.

**Learning Outcomes**

- The student will be able to analyze, design and implement software of reasonable (e.g. up to 3–4000 lines) complexity in the Java programming language. The student will be able to employ suitable advanced programming techniques in this implementation.

- The student will be able to examine a problem specification and write an object-oriented program design for the problem.

- The student will be able to plan the implementation of the program and manage their time to ensure each phase of the implementation is given sufficient attention.

- The student will be able to write documentation for a software project using a standard technique (e.g. javadoc).

- The student will be able to employ standard testing techniques (e.g. unit, integration, system tests; black and white box testing) to ensure the quality of their software.
CHAPTER 2. JUNIOR SOPHISTER YEAR

Content of Course

- Review of Object-Oriented design techniques
- Expressing program design with UML class diagrams
- Some basic principles of project management
- Principles of programming in the Java language
- High level program design in the Java programming language (classes, interfaces, inheritance)
- File and Console IO in Java
- Robust software implementation, exceptions
- Principles of user-interface implementation in Java
- Event driven programming, Java listeners
- Basic threading and concurrency in Java
- Algorithms and data-structures (as required by the project, e.g. $\alpha - \beta$ tree search, backtracking, etc.)
- Software documentation
- Design and application of software testing

Teaching Strategies The teaching strategy is primarily a hands-on practical one, with the students expected to engage in considerable informal discussion and interaction, particularly during the design and early implementation phase of each project. Tutorials are run as informal lectures with question-and-answer and discussion sessions. In the early part of the course lab sessions are run as practical sessions during which students design, write and debug small programs in the Java programming language. During the rest of the semester lab sessions are used to demonstrate progress towards specific milestones on the project implementation, and as discussion and help sessions.
Assessment  Assessment is entirely by coursework. Each project contributes 50% of the end-of-year mark (CD-Stream students have the first project mark combined with the 3C6/B mark).

During the semester the students are expected to demonstrate suitable progress towards milestones in the project. These demonstrations occur in the third, sixth and eighth week labs. Labs are run by postgraduate students—demonstrators—who are expert in the subject, and by the course lecturer. Progress towards these milestones are not formally included in the students final marks.

In the examination of the submitted project marks are primarily awarded for correctness and completeness of implementation with respect to the problem specification, with smaller components (usually around 15%–20% each) for quality of technical documentation and employment of suitable high-level design.

Recommended Texts  Any Java programming textbook, such as by Deitel & Deitel, Smith, etc.
Any software project management textbook, such as Stiller & LeBlanc, etc.

Further Information
Web site:
http://www.tcd.ie/Engineering/Courses/BAI/JS_Subjects/3D5/.
Web site:

2.5 Laboratory and Project Design and Implementation Details

In the Computer Science department, laboratories and practicals are associated directly with lectures and are organised by the lecturers themselves. In the Electronic and Electrical Engineering department, laboratories are centrally organised, and are coordinated by Olga Panarina, olga@mee.tcd.ie.
### 2.5.1 Programme

#### C Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time of Year</th>
<th>Approximate Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C6A Analogue Design Project</td>
<td>Michaelmas Term</td>
<td>24</td>
</tr>
<tr>
<td>3C6B Digital Design Project</td>
<td>Hilary Term</td>
<td>24</td>
</tr>
<tr>
<td>3D1 laboratories</td>
<td>Semester 1</td>
<td>18</td>
</tr>
<tr>
<td>EE laboratories</td>
<td>All Year</td>
<td>30</td>
</tr>
</tbody>
</table>

#### CD Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time of Year</th>
<th>Approximate Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D5A Java Design Project</td>
<td>Michaelmas Term</td>
<td>27</td>
</tr>
<tr>
<td>3C6B Circuit/Systems Design Project</td>
<td>Hilary Term</td>
<td>24</td>
</tr>
<tr>
<td>3D1 Laboratories</td>
<td>Semester 1</td>
<td>18</td>
</tr>
<tr>
<td>3D2 Design Project</td>
<td>Trinity Term</td>
<td>18</td>
</tr>
<tr>
<td>3D3 Laboratories</td>
<td>Semester 2</td>
<td>18</td>
</tr>
<tr>
<td>EE Laboratories</td>
<td>Semester 1</td>
<td>18</td>
</tr>
</tbody>
</table>

#### D Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time of Year</th>
<th>Approximate Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D5A Java Design Project</td>
<td>Michaelmas Term</td>
<td>27</td>
</tr>
<tr>
<td>3D5B CS Design Project B</td>
<td>Hilary Term</td>
<td>27</td>
</tr>
<tr>
<td>3D1 Laboratories</td>
<td>Semester 1</td>
<td>18</td>
</tr>
<tr>
<td>3D2 Design Project</td>
<td>Trinity Term</td>
<td>18</td>
</tr>
<tr>
<td>3D3 Laboratories</td>
<td>Semester 2</td>
<td>18</td>
</tr>
<tr>
<td>3D4 Laboratories</td>
<td>Semester 2</td>
<td>14</td>
</tr>
<tr>
<td>EE Laboratories</td>
<td>Semester 1</td>
<td>12</td>
</tr>
</tbody>
</table>

Note: The CD and D hours do not include time to be spent preparing material for Laboratories or the Project 2.
2.5.2 Timetable

C Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time</th>
<th>Time of Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>3C6A Analogue Design Project</td>
<td>Friday 2-5</td>
<td>Michaelmas Term</td>
</tr>
<tr>
<td>3C6B Digital Design Project</td>
<td>Friday 2-5</td>
<td>Hilary Term</td>
</tr>
<tr>
<td>3D1 Laboratory</td>
<td>Wed. 1-3/3-5</td>
<td>Semester 1</td>
</tr>
<tr>
<td>EE Laboratory</td>
<td>Tues. 2-5</td>
<td>All Year</td>
</tr>
</tbody>
</table>

CD Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time</th>
<th>Time of Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D5A Java Design Project</td>
<td>Friday 2-5</td>
<td>Michaelmas Term</td>
</tr>
<tr>
<td>3C6B Circuit/Systems Design Project</td>
<td>Friday 2-5</td>
<td>Hilary Term</td>
</tr>
<tr>
<td>3D2 Design Project</td>
<td>Fri. 2-5</td>
<td>Trinity Term</td>
</tr>
<tr>
<td>3D1 Laboratory</td>
<td>Wed. 1-3/3-5</td>
<td>Semester 1</td>
</tr>
<tr>
<td>3D3 Laboratory</td>
<td>Wed. 2-4</td>
<td>Semester 2</td>
</tr>
<tr>
<td>EE Laboratory</td>
<td>Tues. 2-5</td>
<td>All Year</td>
</tr>
</tbody>
</table>

D Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Time</th>
<th>Time of Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D5A Java Design Project</td>
<td>Friday 2-5</td>
<td>Michaelmas Term</td>
</tr>
<tr>
<td>3D5B CS Design Project</td>
<td>Friday 2-5</td>
<td>Hilary Term</td>
</tr>
<tr>
<td>3D2 Design Project</td>
<td>Friday 2-5</td>
<td>Trinity Term</td>
</tr>
<tr>
<td>3D1 Laboratory</td>
<td>Wed. 1-3/3-5</td>
<td>Semester 1</td>
</tr>
<tr>
<td>3D3 Laboratory</td>
<td>Wed. 2-4</td>
<td>Semester 2</td>
</tr>
<tr>
<td>3D4 Laboratory</td>
<td>To be advised</td>
<td>Semester 2</td>
</tr>
<tr>
<td>EE Laboratory</td>
<td>Tues. 2-5</td>
<td>Semester 1</td>
</tr>
</tbody>
</table>
2.5.3 Assessment

C Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D1 Laboratories</td>
<td>Up to 20% to 3D1</td>
</tr>
<tr>
<td>EE Laboratories</td>
<td>15% to 3C1, 3C2, 3C3, 3C4 &amp; 3C5</td>
</tr>
</tbody>
</table>

CD Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D1 Laboratories</td>
<td>Up to 20% to 3D1</td>
</tr>
<tr>
<td>3D2 Laboratories</td>
<td>20% to 3D2</td>
</tr>
<tr>
<td>3D3 Laboratories</td>
<td>20% to 3D3</td>
</tr>
<tr>
<td>EE Laboratories</td>
<td>15% to 3C1, 3C2 &amp; 3C4</td>
</tr>
</tbody>
</table>

D Stream

<table>
<thead>
<tr>
<th>Item</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D1 Laboratories</td>
<td>Up to 20% to 3D1</td>
</tr>
<tr>
<td>3D2 Laboratories</td>
<td>20% to 3D2</td>
</tr>
<tr>
<td>3D3 Laboratories</td>
<td>20% to 3D3</td>
</tr>
<tr>
<td>3D4 Laboratories</td>
<td>20% to 3D4</td>
</tr>
<tr>
<td>EE Laboratories</td>
<td>10% to 3C1 &amp; 3C2</td>
</tr>
</tbody>
</table>

2.5.4 Rules

1. We will issue non-satisfactory performance notices (NS Certificates, see Calendar G4 §21,22) as necessary. An attendance roster will be taken for all Laboratories and Project work. It is the student’s responsibility to make sure that they sign in for each session.

   • NS Certificates will be issued at the start of the 2nd term and at the start of the 3rd term to students who either did not attend Laboratory or Project sessions, or who did not show a satisfactory level of participation in the set exercises in the previous term.

   • Students having two (2) NS Certificates may NOT be eligible to sit the end of year examinations.
2. At the start of term 2 we will list the student membership of the various streams C, D and CD. This is the list that we will use to compile results. It is YOUR responsibility to notify us of errors in this list.

3. It is very important that you notify Michael Slevin (the Engineering School Administrator) if you change streams. There are important repercussions for the marking of your project if you fail to do this.
Chapter 3

Senior Sophister Year

3.1 Overall Structure

The overall structure of the streams C, CD and D is as follows: two core subject—4E1 Engineer, Management and Society, 4E2 Project— are complemented by four specialist subjects.

3.1.1 Specialist Courses

Specialist courses generally comprise three lectures and one tutorial per week for. Courses originated by the CS department courses also have practicals associated with them.

Specialist Courses by Stream

The selection of specialist subjects is different for each stream.

- For the C stream, students must take the following pairs of subjects 4C2 with 4C7, 4C4 with 4C10 and 4C5 with 4C3. In addition they must also take one of the following combinations, 4C6 with 4C11 or 4S1.

- For the CD stream, students must take four of the following subject/stream (note: at least one of them must be 4D1 or 4D3): 4D1, 4D3, 4C2 with 4C7, 4C4 with 4C10, 4C5 with 4C3, 4S1.

- For the D stream, students must take 4D1, 4D2, 4D3, and one of 4D4 or 4C2 with 4C7 or 4S1.
3.2 Timetable

Figure 3.1: Senior Sophister Timetable - Michaelmas Term

The official timetable is published on the notice board in the museum building. Up-to-date timetables, including extra details of laboratory arrangements, are maintained on the web at:
http://www.tcd.ie/engineering/Courses/BAI/Joint_CS_EE_SS/Timetables/michaelmas.html and
http://www.tcd.ie/engineering/Courses/BAI/Joint_CS_EE_SS/Timetables/hilary.html
Figure 3.2: Senior Sophister Timetable - Hilary Term
3.3 Course Details

3.3.1 Management for Engineers (4E1)

Lecturers  Frank Bannister and Linda Doyle.

Course Organisation  As is normal for final-year courses, 4E1 runs for the first two terms. It is divided into two components: a project management module in the first term, given by Dr Bannister, and a module on technology and society in the second term, given by Dr Doyle. Each part will be described separately.

Part 1—Project Management

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>Hours of Associated Practical Sessions</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Total</th>
<th>Tutorials Per Week</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>2</td>
<td>18</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Total Contact Hours: 18 45-minute lectures = 13.5 hours

This part of the course is taught in eighteen 45-minute lectures in the Michaelmas term. Students are given exercises to do, but there are no formal tutorials.

Course Description  This part of the course is designed to give students a good grounding in project management. It starts with the rationale for project management, the concept of project definition and project specification. It next covers organisational aspects of projects, estimating, evaluation, costing and project planning. It then covers work breakdown structures, responsibility matrices and a variety of planning tools including critical path method and PERT. The final part of this part of the course covers project management software, project procurement, risk management, cost control and legal aspects of project management.

Learning Outcomes  At the end of part one of the course, student will:

- be able to describe all the stages in a project from conception to completion;
- know how to prepare a detailed project plan;
• know how to use a range of project planning tools and techniques including CPA and PERT.

• understand the nature of project control including risk management, variance analysis and earned value analysis;

• be aware of how projects are organised in different contexts and the roles and responsibilities of the project manager and other parties;

• be able to discuss why projects go wrong and ways to avoid common causes of project failure.

Content of Course

• The importance of project management;

• Why projects fail;

• Project definition;

• Organisation and people management;

• Project feasibility;

• Financial evaluation of projects;

• Work breakdown structures;

• Estimating;

• Quality planning;

• Detailed project specification;

• Costing techniques;

• Absorption and activity based costing;

• Project planning;

• CPA and PERT;

• Project management software;
• Purchasing and procurement;
• Running a project;
• Control techniques;
• Financial reporting;
• Earned value analysis;
• Risk management;
• Legal issues and contract;
• Project closure.

Teaching Strategies  This part of the course is current taught by lecture only. Students are given exercises to try in their own time. The teaching approach combines theory with practical examples from different engineering disciplines.

Assessment  Assessment is by examination.

Recommended Texts  The following is the core text:


The following texts are also referenced:


• Chatfield, C. (2003), Microsoft Project 2003; Step by Step, Microsoft Press International.

The following books are referenced for specific topics:

3.3.  COURSE DETAILS


**Part 2—Technology and Society**

Part 2 of 4E1 runs in the Hilary term. There are two two-hour lectures per week.

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>Hours of Associated Practical Sessions</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Total</th>
<th>Tutorials Per Week</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>0</td>
<td>18</td>
<td>4</td>
<td>40</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Course Description**  The focus of this part of the course is on technology and society. It uses the fields of mobile communications and networking as case studies for exploring issues relating to the design and development of technology in these fields. In particular business, economic, social, ethical, political and other factors that affect the adoption/growth/decline of a technology are studied.

**Learning Outcomes**

- The student will be able to analyse current communication trends from business, economic, social, ethical and political viewpoints.

- The student will be able to debate, challenge and question the value and role of various communication technologies and developments.
• The student will be able to speculate and reason about future trends in the mobile communications field using arguments based not purely on technical issues.

• The student will be able to source material on technological issues that go beyond the technical details of the systems involved.

• The student will be able to discuss and express technical issues in terms suited to a wider non-technical audience.

Content of Course  
The course content cannot be described in a traditional manner as the content is dynamic and changes. However the course is well structured. The course consists of a number of well-defined core lectures presented by the lecturer. These are supplemented by guest lectures that focus on topics of interest. The content of the remaining lecture slots are determined by the students who present lectures themselves on relevant topics using a variety of styles for presentation. The structure is described here with illustrative content as it is the best means for getting a sense of the content.

Core Lectures  
• An overview of a communications technology that is currently in vogue (e.g. 3G mobile communications, WiMax, 802.11)

• A case study of a communications technology that did not succeed/ is appearing not to succeed/ whose success is debatable (e.g. 3G, WAP)

• A focus on some aspects of communication technology that is more future-oriented (e.g. cognitive radio, spectrum trading)

• Communications and society—an exploration of ubiquitous and ambient technology and its role in our lives

Guest Lectures  
The guest lectures are chosen from different walks of life to discuss the various topics of relevance. The course has been running in its present form for two years so some of the guest lectures are listed here:

Business Lectures (Denis O’Brien (Entrepreneur), Sean O’Sullivan (Rococco— mobile communications company)
3.3. COURSE DETAILS

- Political Lectures (Eamonn Ryan (Green Party spokesperson on Communications), Simon Coveney (Fine Gael spokesperson on Communications)
- Access Issues Lectures (Alexis Donnelly—lecture on designing for disability)
- Public Sector Lectures (Richard Horton (Comreg), Brendan Tuohy (General Secretary—Department of Communications)
- Society, Art & Technology Lectures (Jonah Brucker-Cohen & Katherine Moriwaki, Artists and technoligists)
- Research and Communications (Prof. Donal O’Mahony (Networks & Telecommunications Research Group))
- Professionla Body Lectures (IEE representative lecture)

**Student Lectures**

- Contributions from groups of students typically taking the form of making the case in favour of certain technologies are presenting findings from reports and papers on communications and society in general.

**Contribution to Programme**

<table>
<thead>
<tr>
<th>Science and Mathematics</th>
<th>Discipline Specific Technology</th>
<th>Information and Communications Technology</th>
<th>Design and Development</th>
<th>Engineering Practice</th>
<th>Social and Business Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>H</td>
<td>-</td>
<td>-</td>
<td>H</td>
</tr>
</tbody>
</table>

**Contribution to IEI Programme Areas**

The course contributes to the IEI Programme Outcomes as follows:

<table>
<thead>
<tr>
<th>(a) The ability to derive and apply solutions from a knowledge of sciences, engineering sciences, technology and mathematics</th>
<th>(b) The ability to identify, formulate, analyse and solve engineering problems</th>
<th>(c) The ability to design a system, component or process to meet specified needs</th>
<th>(d) An understanding of the need for high ethical standards in the practice of engineering, …</th>
<th>(e) The ability to work effectively as an individual, in teams and in multidisciplinary settings …</th>
<th>(f) The ability to communicate effectively with the engineering community and with society at large</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>M</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>
Contribution to IEI Programme Outcomes

**Teaching Strategies**  The teaching strategy for is all about getting students to think outside the box and to express their thoughts in a comprehensive, clear and accessible manner. This is achieved both by introducing the students to a wide range of material through core lectures and invited lectures as well as getting students to take the initiative and create content themselves. In terms of creating content time is set aside for research and design of the content. These activities are performed in groups (e.g. in computer labs where internet access is available) but all students get the opportunity to present as well as question and evaluate other presentations. A very large emphasis is placed on students giving their opinion and on questioning issues and ideas. The overall philosophy on the course is to encourage as much interaction and debate as possible.

**Assessment**  The students are assessed through a mixture of continuous assessment and examination. Continuous assessment happens throughout this part of the course. The students are given tasks during each lecture that range from simple tasks such as summarising critical points made by guest lectures to more complex tasks that involve designing a lecture themselves. 40% of the marks for this part of the course (i.e. 20% of the overall 4E1 result) are obtained through continuous assessment. This part of the course is also examined during the annual examinations. It consists of 50% of the 4E1 examination. There is one compulsory open-ended question on the exam.

**Recommended Texts**  Everything that is relevant from all types of media.

**Further Information for 4E1**  ECTS Credits: 10.

### 3.3.2 Final Year Project (4E2)

**Coordinators**  Professor J. K. Vij (EEE) and Dr. K Dawson-Howe (CS)

**Organisation**  As part of the fulfilment of the final year course of the BAI, the students in C and CD streams are required to carry out an individual engineering project. To this end, each student is assigned a project topic and
supervisor who will guide the course of the project throughout the academic year. There are no formal timetabled hours associated with the project but students are expected to spend the time it takes to make reasonable progress and to keep in regular contact with their supervisors. It is recommended that students make a formal arrangement with their supervisors to meet on a weekly basis, preferably at a regular appointed time.

There are two separate coordinators of projects in each department. For 2005/06 in EEE it is Prof. J. Vij and in CS it is Dr. K. Dawson-Howe. Students may opt for projects supervised by lecturers in any of EEE or CS Departments, provided the chosen topic is judged suitable by the coordinators. Projects in the EEE Department are assigned to students once students have selected a broad topic area of preference, and lecturers have provided project descriptions on the form shown below. In the CS Department lecturers advertise projects and are approached by students wishing to take them on.

Objectives

- To give students the opportunity to put into personal practice the knowledge and skills acquired throughout the course of the BAI.
- To allow students to gain experience of independent enquiry and investigation of a practical engineering problem, application or topic.
- To facilitate the development of an ability to assess and criticise information, methods and results for a defined engineering purpose.
- To develop the students personal skills in relation to project management and technical writing and expression.

Learning Outcomes  The engineering project is designed such that at the completion of the exercise the student will be able to

- Design a system, component or process to meet a specified goal
- Communicate effectively in technical and scientific writing, and to present scientific/technical ideas concisely to a technical audience that may not be expert in the specific domain of the presentation
• Identify and formulate technical problems in such a manner as to make them amenable to solution

• Analyse and interpret results from experiments conducted during the course of the design process in order to modify improve or explain the functionality of the system, component or process being created

• Formulate the design of systems in terms of a schedule of intermediate goals that manifest in subsystems

• Manage workflow and task scheduling within the constraints of resources and time given specific design goals and deadlines

• Use industry standard hardware and/or software tools for all aspects of design including analysis and presentation

• Derive, apply and adapt solutions from the discipline specific knowledge gained in coursework, to a real world problem solving context

Assessment  The project counts for 120 of the total 600 marks awarded at Senior Sophister annual examinations. Assessment is governed by a set of overall guidelines (see page ??). The exact procedures very slightly between the departments:

Electronic & Electrical —Engineering  The project is assessed by means of several written submissions which will be read by both the project supervisor and an assigned second reader. The submissions are as follows:

• Definition (5 marks): A one-page outline of the project giving details of the nature of the project, the work the student intends to carry out and a target timetable for accomplishing the major tasks involved.

• Summary (15 marks): A 5-page summary will be required towards the end of Hilary term, which describes briefly the main aims and objectives of the project, the work which was carried out by the student and the results obtained as well as a conclusion on the outcome.

• Main Report (100 marks): A full written report on the project, which should not exceed 50 pages. This report should be properly structured and typed in accordance with instructions given by the department. The report should contain an introduction outlining in detail
3.3. COURSE DETAILS

the aims and objectives of, as well as some background information on the project. The bulk of the report should discuss in detail the main technical work carried out by the student with appropriate results, assessments and deductions. A final conclusion should comment on the overall outcome of the project.

There is also usually a short interview held with the external examiner. In the academic year 2003-2004, each student gave poster presentation to the External, the Internal Examiners and to the Project supervisor. This presentation would normally take place immediately after the end of Hilary term or in Trinity term. The posters were presented to the entire junior Sophister class.

The project marking sheet used by the Electronic & Electrical Engineering department is on page ??.

Computer Science  The project is assessed on the basis of the final year report. The project supervisor and the second reader attend a project demonstration held early in the Trinity term. The demonstration is informal and is not marked. The project marking sheet used by the Computer Science department is on page ??.

The project marking sheet used by the first and second readers is attached.

3.3.3 Microelectronic Circuits 1/2 (4C2/4C7)

Lecturers:  Dr J.B. Foley, Dr M.J. Burke.

Organisation:  3 lectures and 1 tutorial per week; 4C2 Michaelmas term; 4C7 Hilary term.

Assessment:  A single 3-hour examination paper will account for 80% of overall assessment of the subject 4C2/4C7. The examination paper will contain 8 questions and will be divided into two sections, each of 4 questions, with a requirement to attempt 5 questions, including at least two questions from each section. The remaining 20% of the overall assessment mark will derive from two written assignments to be carried out and submitted during the course of the year, 10% from a 4C2 assignment and 10% from a 4C7
CHAPTER 3. SENIOR SOPHISTER YEAR

assignment. Both assignments must be handed and signed in at the Department Office, Dept. of Electronic and Electrical Engineering, Printing House on dates which will be specified by the lecturers concerned.

4C2 Microelectronic Circuits I (Michaelmas Term)

Objectives:

To develop an electrical understanding of the metal-oxide-semiconductor (MOS) field effect transistor;

To treat the fundamental static and dynamic performance of simple CMOS circuits noting design trade-offs;

To build up a knowledge of CMOS logic structures;

To extend the knowledge of dynamic performance to more complex logic structures and systems;

To introduce some principles of system design.

Syllabus:

• The MOSFET: physical principles of device operation; current voltage relationships, device models; second order effects.

• Static Circuit Analysis: MOS inverters; the CMOS inverter transfer characteristic and its switching level; NAND and NOR gates; noise margin; transmission gate.

• Dynamic Circuit Analysis: circuit lay-out, MOS transistor capacitances; inverter step response; gate delays; power dissipation.

• CMOS Logic Functions: generalized CMOS combinational logic; XOR and transmission gate logic; sequential logic elements.

• CMOS Subsystem performance: finite input transition times; gate delays; buffer design; interconnect delays.
Reading List:


4C7 Microelectronic Circuits II (Hilary Term)

Objectives:

To develop an understanding of the operation of the MOSFET as a linear amplifying circuit element

To introduce basic analogue sub-circuits which may be used as building blocks for larger circuits

To develop the skills required for the analysis of simple CMOS amplifier structures

To lay the foundation principles of modern CMOS amplifier design

To examine some applications involving the use analogue CMOS circuits.

Syllabus:

- MOSFET as an Amplifying Device: The use of the MOSFET in the saturation region, large signal and small signal models, voltage-controlled current source, transconductance, gain, output impedance.
• CMOS Analogue Sub-circuits: voltage references, current sources, current mirrors.

• Simple Amplifiers: single-stage amplifiers, enhancement loaded, depletion loaded, current-source loaded stages.

• Frequency Response Considerations: sources and effects of parasitic capacitances, frequency response, gain and phase margins.

• Differential Amplifiers: differential input stage, current mirror loaded differential amplifier, performance characteristics.

• Intermediate and Output Stages: source follower, push-pull stages, impedance translation.

• Simple Operational Amplifier Structures: basic low-performance and medium performance operational amplifiers.

• Linear Applications: To be defined.

Reading List:


3.3.4 Telecomms / Digital Communications (4C4/4C10)

Lecturers: Prof. P.C. Fannin and Mr L. Dowling

Organisation: 3 lectures and 1 tutorial per week; 4C4 Michaelmas Term; 4C10 Hilary Term.
3.3. COURSE DETAILS

Assessment: There are three lectures per week and one tutorial per week. Additionally, the students must complete a course of laboratory work, which is compulsory. At the discretion of the course examiners, up to ten percent of the available marks for EE4C42 may be assigned for satisfactory completion of the laboratory course. The remainder of the available marks will be based on the student’s submitted EE4C42 examination script.

Examination: For examination purposes, 4C4 and 4C10 are combined and taken as one, 3 hour exam. Students are requested to answer 5 from 8 questions, including at least 2 from each section.

4C4 Telecommunications (Michaelmas term)

Objectives: To expand on the concepts introduced in the third year course (3C5) on Telecommunications. To enable the student, to understand antenna and propagation mechanisms together with digital modulation and demodulation techniques and to appreciate the constraints imposed on a communications channel in terms of data and error rates. This course establishes analysis tools and concepts of modern digital communications and information theory. This material should be considered a prerequisite for anybody wishing to pursue careers in Communications and Information Networking related areas.

Syllabus:

- Basic antenna engineering
- Modulation
- Noise, information and channel capacity
- Baseband pulse transmission and line codes
- Cross-talk Probability of error in AWGN channel
- Correlation techniques and signal recovery
- Laboratory work Labs: 3 hour duration
- Pulse amplitude Modulation
• Signal synthesis
• The division multiplexing
• Delta Modulation
• Aerial measurements
• Slotted line measurements

Reading List:


Staelin et al. Electromagnetic Waves, Prentice Hall


4C10 Digital Communications (Hilary term)

Objectives: The purpose of this course is to provide the mathematical methods and present the theoretical models which are necessary to understand the operation of modern digital communications systems and source coding algorithms.
3.3. COURSE DETAILS

Syllabus:

- Error control coding
- Signals and Systems
- Modulation, demodulation and waveform design
- Equalisation
- Multiple Access Techniques
- Laboratory Work

Reading List:


3.3.5 Digital Signal Processing / Digital Control Systems (4C5/4C3)

Lecturers: Prof. F.M. Boland and Dr A. Quinn.
**Organisation:** 3 lectures and 1 tutorial per week; 4C5 Michaelmas term; 4C3 Hilary term

**Assessment:** One 3-hour end-of-year exam on a combined paper will account for 70 marks out of the total of 100 marks for 4C5/8. Working in groups, students are required to contribute to an investigation of an application of DSP. A typical project requires the use of Matlab for signal analysis. The project is conducted during timetabled laboratory periods and each group must make a presentation of their results. The project is assessed based on the quality of the presentation and on an individual written report from each student. The project accounts for 15 marks out of the total of 100 marks for 4C5/8. The course will be supplemented by about four software-based laboratories, which must be attended by all students. In addition, a number of written assignments (at the discretion of the lecturer), based on the 4C8 lectures and laboratories, will be required from students at intervals throughout the course. This work will be marked, and constitute 15 marks out of the total of 100 marks for 4C5/8.

**4C5 Digital Signal Processing I (Michaelmas Term)**

**Objectives:** Digital Signal Processing (DSP) is concerned with the processing of signals which are represented as sequences of finite-precision numbers. This course, 4C5- DSP I, is an introduction to the theory and applications of digital signal processing. Experimental DSP projects are an integral part of the course and support the technical material covered in the lectures.

**Syllabus:**

- Discrete time sequences: Basic classification of sequences, elementary sample rate conversion

- Discrete time systems: (a) Time Domain Classification of systems—linear/nonlinear, causal; Convolution; BIBO stable; FIR, IIR systems; Deconvolution; Block diagrams. (b) Frequency Domain, Frequency response of BIBO stable discrete time, linear, shift invariant systems, The Discrete Time Fourier Transform.

- Sampling and reconstruction: Sampling and quantising continuous time signals, Data Holds, signal reconstruction.
3.3. COURSE DETAILS

- Filter functions in continuous time.
- The ideal low pass filter, approximations, frequency transformations.
- Digital Filters in the z-domain: Transfer functions, Pole-zero maps, Frequency response, IIR filter design using bilinear transform, FIR filter design using windows.

Reading List:

Edward P. Cunningham, Digital Filtering: An Introduction, Houghton Mifflin
Lonnie C Lundeman, Fundamentals of Digital Signal Processing, Wiley
Richard A. Roberts and Clifford T. Mullis, Digital Signal Processing, Addison-Wesley
B. Mulgrew, P. Grant & J. Thompson, Digital Signal Processing Concepts and Applications
Boaz Porat, A Course in Digital Signal Processing, John Wiley and Sons
Kenneth Steiglitz, A DSP primer: with Applications to Digital Audio and Computer Music, Addison Wesley

4C3 Digital Signal Processing II (Hilary Term)

Objectives: 4C8 is a course of further Digital Signal Processing, extending the material presented in DSP I (4C5). Techniques in the design and analysis of linear, time-invariant discrete systems are revised and extended, with emphasis on such topics as rapid system prototyping, frequency response sketching, generalized linear phase filter design, etc. Computational structures are presented, and their algorithmic properties explored. The Discrete
Fourier Transform (DFT) and its rapid computation (the Fast Fourier Transform (FFT)) are studied, so that the student can understand how frequency spectra are actually calculated using digital computers. A key objective of the course is to extend the analysis framework for deterministic signals to the world of information signals (statistical signal processing). The aim here is to provide the student with an understanding, firmly rooted in basic DSP theory, of modern DSP algorithms. Emphasis is placed on the Linear Predictive Coding (LPC) methodology for modelling, prediction and compression of speech. An introduction to adaptive Wiener filtering for non-stationary information signals is also provided. The course material is supported by laboratories on selected topics in Matlab.

**Syllabus:**

- Characterization methods for discrete-time systems (review).
- Realizations (i.e. computational structures) for discrete-time systems.
- State-space analysis of discrete-time systems.
- The Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT).
- The theory of discrete-time random processes.
- Modelling, prediction and filtering of discrete-time random processes.
- Adaptive discrete-time signal processing.
- Applications in speech analysis.

**Reading List:**


3.3. COURSE DETAILS


3.3.6 4C6/4C11 Microelectronic Technology and Optoelectronics

Lecturers: Dr R.A. Moore, Prof. E. McCabe, Dr J. Donegan.

4C6 Microelectronic Technology (Michaelmas Term)

Organisation: 3 hours of lectures and 1 tutorial per week; 4C6 Michaelmas Term; 4C11 Hilary Term.

Assessment: There will be a single 2-hour examination paper. The paper will contain 6 questions with a requirement to answer 4.

Objectives: To gain in-depth knowledge of the processing techniques used to fabricate an integrated circuit (IC) and how these processes are applied to the production of both bipolar and MOS ICs. This is a practical course covering the basics of IC fabrication. It will be of benefit to those seeking to pursue a career in areas such as IC Design and IC Technology.

Syllabus:

- Processing techniques: The following techniques will be examined in some detail: crystal growth; diffusion; ion-implantation; oxidation; lithography; metallisation; plasma etching. There will be a guest speaker from industry to give an industrial perspective.

- Spectroscopy: Introductory material on Raman and FTIR spectroscopy will be presented.
• Devices/Circuits: Techniques for fabrication of the following devices, using the processing techniques described above, will be studied:

• Bipolar: discrete transistor; monolithic IC transistor; resistor; diode; capacitor; circuit layout for ECL NOR gate.

• MOS: PMOS, NMOS and CMOS technologies will be introduced and the problem of latch-up will be discussed.

• Silicon-on-Insulator will be proposed as an alternative to bulk silicon.

Labs—3-hour duration: Laboratories will consist of hands-on experience in fabricating a simple IC incorporating the processing steps leading to the fabrication and test of a simple integrated circuit.

Reading List:

Recommended Text:


Further Reading:

SM Sze, Semicinductor Devices: Physics and Technology, Wiley,1985


4C11 Optoelectronics (Hilary Term)

Organisation: 3 lectures + 1 tutorial per week

Assessment: There will be a single 2.5 hour examination paper. The paper will contain 6 questions with a requirement to answer 4.
3.3. COURSE DETAILS

Objectives: This course focuses on the interaction of light and electronic materials. It brings together the students basic knowledge of electronic devices and optics in a two-strand format. The first deals with the more detailed understanding of the light-semiconductor interaction introducing the exciting area of quantum semiconductor devices, for example. The second considers the applications of these interactions in an engineering context. Much of the attention in this second strand is focused on the multi-faceted telecommunications applications. Optoelectronics has revolutionised the telecommunications industry and has enormous scientific and commercial potential in other industries. This course provides a background which should give students the opportunity to come to terms with and ultimately exploit the technology continuously coming on-stream in this fast moving field.

Syllabus:

- Semiconductor laser physics.
- Quantum semiconductor devices.
- Non-linear interactions—e.g. solutions in communications systems.
- Electro-optic effect.
- Semiconductor light detectors.
- Optical waveguides—particularly optical fibre waveguides.
- Optical systems—how to put light sources, detectors and waveguides together to make a feasible telecommunications system.

Reading List:


Claude Weisbuch and Borge Vinter, Quantum semiconductor Structures,

Ghatak and Thagarajan, Optical Electronics,
3.3.7 Integrated Systems Design (4S1)

Lecturer  John McCarthy.

Course Organization

<table>
<thead>
<tr>
<th>Engineering</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures</th>
<th>Tutorials</th>
<th>Practicals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester or Term</td>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
</tr>
<tr>
<td>1,2</td>
<td>3</td>
<td>54</td>
<td>1</td>
<td>11</td>
<td>2</td>
</tr>
</tbody>
</table>

Total Contact Hours: 90

Course Description, Aims and Contribution to Programme  Integrated System Design is a two-semester course taken by Senior Sophister C, CS, and CD stream students. It is divided into two, non-exclusive parts, namely Verilog HDL and Design Fundamentals.

Verilog HDL is an IEEE recognized hardware language used in the design of digital systems. This part of the course introduces the students to the fundamentals of hardware languages. The students then gain knowledge of digital design techniques and learn how to apply Verilog language constructs in order to tackle any hardware problem. Implementation at a behavioral level is first analyzed, after which the student migrates to the RTL and gate level netlists. Complex state machines and data-transfer blocks are constructed as part of the term work.

Design Fundamentals concentrates on the design flow used within chip and system design. It investigates the practices within industry and complements the material learnt in the Verilog part of the course. Integrated systems are highly complex, where both design and subsequent testing are controlled by strategies developed within the remit of the design flow. This course introduces the student to such complex systems and allows them to develop in order to tackle problems head on. The stages within the design flow, such as synthesis and test are considered in flow, with considerable attention paid to marketplace strategies.

Learning Outcomes  On completion of this course the student will be able

- Develop understanding of the methods necessary to create complex integrated systems
3.3. COURSE DETAILS

- Create a fully functional rtl level code that mimics design specification
- Decide on the necessary partitioning within synthesis and the associated trade-offs of their decisions
- Analyse scan insertion and the advantages and disadvantages of such an approach
- Create data transfer protocol blocks at a system level
- Analyse the problems of clock tree distribution and power hungry nets and decide on the appropriate architecture
- Implement test strategies at key milestones within the design roadmap

Content of Course

- Verilog HDL
  - Use of gate level primitives to create a functional netlist
  - If-else and case statements and the resulting hardware network
  - For loops, while loops and initial statements
  - Procedural statements
  - Blocking/non-blocking assignments
  - Timing Control, Delays, SDF files
  - Submodule instantiation, declarations, port mapping
  - State Machines: Moore and Mealy machines
  - One hot or Binary Encoding different techniques. Using FPGAs for one hot encoding
  - Single or multiple process machines
  - Task and function procedures
  - Conditional and unconditional statements
  - Synchronous devices with asynchronous global resets
  - Synthesis coding techniques
  - Detailed testbenches with indexed sub-blocks
— fork and join statements
— Switch Level and PLI Verilog
— 8x8 Fifo
— USB block

• Design Fundamentals

— Historical perspective of system design; Development in the last 40 years; Socio-economic benefits and future needs
— Design flow diagram and related tangents
— Key design parameters and associated tradeoffs.
— Clocking strategies within a design, power coupling on nets, clock tree insertion; layout issues between front and back end integration
— Asynchronous data transfer, handshaking, timing analysis and benchmarking
— Synthesis strategies, i.e., top down or bottom up; partitioning rules; tcl script generation for automated synthesis; post compile timing analysis, input/output timing criterion
— Scan insertion and related overhead in hardware; controllability and observability of generated vector suites. Vector compression algorithms and associated market dynamics
— Review of system level data transfer techniques, and the link to verilog coding;
— Important hardware /software interface issues;
— Portability of application and reusability; the succession of IP into ASIC design
— Usage of ATPG for scan and JTAG for boundary scan insertion, as well as the historical perspective of such an implementation
— Key models in data paths and the critical paths of various implementations
— Key sub-module analyses, focusing on processing techniques and the strategies for finding faults; system evaluation protocols
Teaching Strategies  The strategy is a mixture of lectures, problem solving laboratories and interactive class demonstrations. For the first part of the course, formal lectures are accompanied by notes containing gaps that student fill in as they attend the lectures. The notes are similar to workbooks in parts and students have to complete various sections during these lectures. During tutorials, students work on design problems with the aid of the lecturer. Material covered in classes preceding the tutorial are examined and questions relating to this are posed by the lecturer. Once a realistic attempt has been attempted by the class, the lecturer then proceeds to go through the solution piecemeal, encouraging the students to actively engage in questioning the proposed solutions and its merits based on material presented in class.

Laboratory work is particularly related to the course work of 4S1/4BA11. These sessions are divided into two sub sections. The initial section takes place in the first term, where students gain familiarity with a working system and board while performing activities such as coding and synthesis as presented in the course work. Laboratory sessions in the second term see the class divide into teams where work is carried out on separate projects. These projects encourage the student to go through the whole design flow, from specification to creation, and understand the various stages along the way.

The lectures are punctuated by interactive sessions with the students, where case studies are presented and discussed. This allows the class the opportunity to learn about effective oral communication and provides a way of integrating effective teaching with an enjoyable atmosphere.

Assessment  Exam work counts for 85% of overall mark. 5% is given to the first section of individual laboratories, while 15% is given to the projects set in the second term.

Recommended Texts

CHAPTER 3. SENIOR SOPHISTER YEAR


Further Information
ECTS Credits: 10.

3.3.8 Networking and Advanced Microprocessor Systems (4D1)

Lecturers  Hitesh Tewari and Jeremy Jones.

Course Organisation  This course is taught in two parts. The first part [Networking] occupies weeks 1-9 of Michaelmas term while the second [Advanced Microprocessor Systems] occupies weeks 1-9 of Hilary Term.

<table>
<thead>
<tr>
<th>Engineering</th>
<th>Start</th>
<th>Hours of Associated Practical Sessions</th>
<th>End</th>
<th>Lectures</th>
<th>Tutorials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester or Term</td>
<td>Week</td>
<td>Week</td>
<td>Per Week</td>
<td>Total</td>
<td>Per Week</td>
</tr>
<tr>
<td>Term 1</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>Term 2</td>
<td>10</td>
<td>–</td>
<td>18</td>
<td>3</td>
<td>23</td>
</tr>
</tbody>
</table>

Total Contact Hours: 45+27 = 72

Course Description, Aims and Contribution to Programme  Networking is a 9 week module taken by Senior Sophister CD and D Stream students. The course concentrates on building a sound foundation for the understanding Data Communications and Computer Networks. The course structure is based around the OSI Reference Model and deals with the major issues in bottom 4 (Physical, Data Link, Network and Transport) layers of the model. The students are also introduced to the areas of Network Security and Mobile Communications. The student gains enough knowledge to be able to understand the basics of recent developments in the areas of computer networking.

Advanced Microprocessor Systems  This course follows on from 3D2 and focuses on the techniques used by modern microprocessor systems to provide (i) virtual memory and (ii) high performance. Topics covered are virtual memory, caches, multiprocessors, multiprocessor cache coherency, multiprocessor spin locks implementations and main memory subsystems. The aim of the course is to explain (i) how high performance is obtained and (ii) the
close relationship between the hardware and software inherent in the design of modern microprocessor.

Learning Outcomes

Networking

• The student will be able to analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.

• The student will be able to specify, design and test new and modified data communications protocols.

• The student will be able to analyze, specify and design the topological and routing strategies for an IP based networking infrastructure.

• The student will be able to provide an in-depth commentary on future networking and data communications technologies, with a view to incorporating these into the existing organizational network environment.

Advanced Microprocessor Systems  Students will be able to:

• Explain the advantages of using virtual memory, show how virtual addresses are mapped efficiently to physical addresses and demonstrate how the functionality of a MMU is integrated into an operating system.

• Explain the use of a memory hierarchy to reduce effective memory access times, describe the organisation and operation of a cache, evaluate the hit rate of a cache given an address trace, develop a C/C++ program to model a cache and know how to apply address trace analysis optimisations.

• Discuss the problems of using caches in a multiprocessor environment, know the operation of several cache coherency protocols and be able to predict the bus traffic given a sequence of CPU “memory” accesses.

• Know the operation of several multiprocessor spin lock algorithms and be able to evaluate their performance.
• Design a DRAM based error correcting memory subsystem and evaluate the advantages of using VRAMs based graphics subsystems.

Content of Course

Networking

– Introduction
  * Computer Networks
  * Standards Bodies
  * OSI Reference Model

– Physical Layer
  * Theoretical Aspects
  * Encoding Schemes
  * RS-232 Interface
  * PSTN & Modems
  * DSL Technologies
  * Circuit & Packet Switching

– Data Link Layer
  * Asynchronous and Synchronous Transmission
  * Character & Bit Oriented Framing
  * Flow Control
  * Data Compression
  * Error Detection & Correction Techniques
  * Error Control
  * SLIP & PPP

– Local Area Networks
  * Network Topologies
  * IEEE 802.3 - Ethernet
  * IEEE 802.5 Token Ring

– High Speed Networks
  * IEEE 802.3u Fast Ethernet
  * IEEE 802.12z Gigabit Ethernet
3.3. COURSE DETAILS

- LAN Switching
- FDDI
- Network Security
  - Classical Cryptography
  - Symmetric Key Cryptography
  - Public Key Cryptography
  - Authentication and Digital Signatures
  - X.509 Digital Certificates
- Internet Suite of Protocols
  - Internet Protocol (IP)
  - ARP & RARP
  - Routing Issues
  - Addressing & Subnets
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
  - IPv6 Next Generation IP
- Mobile Communications
  - 2G, 3G & Beyond
  - Mobile IP (Macro/Micro Mobility)
  - IEEE 802.11 - Wireless LANs/WiFi
  - Personal Area Networks Bluetooth
  - Mobile Ad Hoc Networks

Advanced Microprocessor Systems

- Virtual Memory.
- Memory Management Units (MMUs).
- Multi-level page tables.
- Translation look aside buffers (TLB).
- MMU integration into an OS kernel.
- Cache organisation (L, K & N) and operation.
- Cache performance.
– Address trace analysis.
– Memory consistency.
– Multiprocessor architectures
– Multiprocessor cache coherency protocols (write-through, write-
one, Firefly and MESI)
– Simple spin lock algorithms.
– Ticket locks.
– Load locked/store conditional based locks.
– Evaluation of spin lock algorithms.
– DRAM interfacing
– Error correcting codes
– VRAMs

Teaching Strategies

**Networking** The teaching strategy is a mixture of lectures and laboratory exercises designed to provide the student with a better knowledge of some of networking protocols. The students attend formal lectures during which they are given handouts of the course notes with a few gaps that they fill in during the course of the lecture. The emphasis during the lecture is on allowing the student to focus on the lecture and pose question on various issues that may arise. The laboratory exercises are designed with a number of factors in mind. The first and foremost is to allow the student to develop a number of data communications simulations such as the design of a Token Ring network, a Data Link protocol or a Web Server etc. The second is to expose the students to a functional, concurrent language such as Erlang. The final aim is to allow the student to develop their programs in a UNIX environment.

**Advanced Microprocessor Systems** The teaching strategy is a mixture of lectures and problem solving tutorials. Students make use of interactive web based animations to explore the operation of caches and cache coherency protocols.
3.3. COURSE DETAILS

Assessment  80% of the assessment is due to a three hour examination held during Trinity Term. The remaining 20% is allocated for practical work divided equally between the Networking and Advanced Microprocessor Systems parts of the course. For the first part of the course, the students are given a total of four laboratory exercises and one substantial project, all of which involve the use of the Erlang Programming language to develop network simulations.

The exam is organised in two parts, with four questions in each part. Students must answer five questions, at least two questions from each part.

Recommended Texts

- *Data Communications, Computer Networks & Open Systems*, 4th Ed., Fred Halsall, Addison Wesley
- *Electronic Payment Systems* 2nd Ed., Donal O’Mahony et al., Artech House
- *Mobile Communications*, Jochen Schiller, Addison Wesley
- *Computer Architecture—a Quantative Approach*, John Hennessey & David Patterson
- *High Performance Computer Architecture*, Harold Stone

Further Information

Web site:  
http://www.cs.tcd.ie/htewari/4D1/

Web site:  

ECTS Credits: 10.
3.3.9 Knowledge and Data Engineering (4D2)

Parts 2 & 3: Symbolic Programming and AI

Lecturer: Professor Pádraig Cunningham Padraig.Cunningham@cs.tcd.ie.

Objective: The objective of this part of the course is to present an introduction to Applications of Artificial Intelligence. Programming in Common Lisp is presented as a means of developing applications that involve symbol manipulation. On completion of this course students should have a good understanding of the special characteristics of symbolic programming. They will understand how basic AI techniques of search can be implemented in software. They will also understand the operation and potential application of modern AI techniques such as Neural Networks, Case-Based Reasoning, etc.

Content: This section of the course comprises two parts: Symbolic Programming in Common Lisp and AI Techniques and Applications.

- Common Lisp
  - Symbolic Programming; Why?, When?
  - Introduction to Common Lisp
  - Symbolic Programming Examples
    * Symbolic Differentiation
    * Planning Problems
  - Search
    * Depth First & Breadth First
    * Hill Climbing & Best First Search
    * Branch & Bound
    * A*
    * Search in simple games

- AI Techniques and Applications
  - Stochastic Search
    * Simulated Annealing
3.3. COURSE DETAILS

* Genetic Algorithms
  - Rules-Based Programming
  - Naive Bayes Classifiers
  - Induction of Decision Trees
  - Neural Networks
    * Feedforward Networks, Error Backpropagation
    * Hopfield Nets
    * Self Organising Feature Maps
    * ART Networks
  - Case-Based Reasoning
    * k-Nearest Neighbour Case Retrieval
    * Case Studies
  - Planning Systems
  - Data Mining
    * Collaborative Recommendation
    * Searching the Web

Primary texts: Most of the primary course material is covered by the following two basic texts, it will be necessary to refer to other reading material which is listed under Auxiliary texts.


Auxiliary texts: All of the secondary texts are accessible in the Library.


*AI* Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

3.3.10 Operating Systems and Distributed Systems (4D3)

Lecturer  Jonathan Dukes.

Course Organization  This course runs for eighteen weeks during Michaelmas and Hilary terms and comprises three lectures and one practical laboratory session each week. There are no practical laboratory sessions during the final six weeks of the course. The total contact time is 54 hours of lectures and 12 hours of supervised practical laboratory work. The course is taken by Senior Sophister CD- and D-Stream Engineering students. (The course is optional for CD-stream students, who must choose at least one of CS4D1 and CS4D3.)

<table>
<thead>
<tr>
<th>Engineering Semester or Term</th>
<th>Start Week</th>
<th>End Week</th>
<th>Lectures Per Week</th>
<th>Total</th>
<th>Practical Per Week</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michelmas Term</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>27</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Hilary Term</td>
<td>10</td>
<td>12</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Hilary Term</td>
<td>13</td>
<td>18</td>
<td>3</td>
<td>18</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Total Contact Hours:</td>
<td></td>
<td></td>
<td></td>
<td>66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Course Description, Aims and Contribution to Programme  Operating Systems and Distributed Systems is a full year course consisting of two parts. The first part of the course, lasting approximately ten weeks, covers both practical and theoretical topics in the area of distributed systems. The second part of the course, lasting approximately eight weeks, covers various aspects of the design of modern operating systems. In both parts of the course, some of the common algorithms, models and techniques used in the areas of Operating Systems and Distributed Systems are presented, along with a number of case studies. Throughout the course, students are encouraged to consider the suitability of an algorithm, model or technique in different circumstances and to compare different approaches to solving problems.

In the area of distributed systems, the aim of the course is to enable students to design, construct and reason about a distributed system of moderate complexity. In the area of operating systems, the aim is to equip students with a knowledge of some of the common algorithms and policies used by modern operating systems, enabling them to assess the suitability of different approaches in a given context. This course also places a strong emphasis on
independent practical work, encouraging students to develop their programming, research and communication skills.

Learning Outcomes

- Students will be able to describe algorithms and architectural models used to implement distributed file systems, physical clock synchronization, logical clocks, elections, mutual exclusion, multicast message ordering, transactions and replication in distributed systems.

- Students will be able to identify the advantages and disadvantages of algorithms and models in the areas identified above and will be able to assess their suitability for a given application, using their knowledge of the challenges presented by distributed systems.

- Students will be able to design, construct and describe a distributed system using a distributed object-oriented programming technology and will be able to appraise a design and identify its limitations.

- Students will be able to describe and discuss an unfamiliar technology in the area of distributed systems and explain how the technology could be employed.

- Students will be able to describe algorithms and techniques used by modern operating systems for memory management, thread scheduling, disk I/O and file management and will also be able to compare different algorithms and techniques and discuss their suitability in a given context.

Content of Course

- Distributed Systems
  - Distributed programming: Java RMI
  - Distributed file systems: NFS and AFS are examined as case studies
  - Physical clock synchronization, logical clocks and vector clocks
  - Multicast message ordering
  - Elections and mutual exclusion in distributed systems
Distributed transactions and the two-phase commit protocol
- Replication: replication models and group communication

- Operating Systems
  - Operating system fundamentals: operating system services and kernel architectures
  - Thread scheduling: thread states, performance metrics, scheduling algorithms and case studies
  - Memory management: overview of virtual memory, page-fault handling, policies and resident set management
  - Disk I/O and scheduling: disk I/O scheduling policies and motivations
  - File systems: case studies including NTFS and UNIX, free block management, buffer caches

Teaching Strategies  Lectures are used to present students with the core course material and interaction during the lectures is encouraged. The students are provided with an electronic copy of the lecture material but are expected to supplement this with worked examples also presented during lectures.

A small number (approximately five) tutorials are held during the year, with each tutorial using either one or two of the lecture contact hours. These tutorials are unmarked and are used instead to encourage further discussion or reinforce the concepts presented during lectures.

Approximately 12 practical laboratory sessions throughout the year allow students to further develop their programming skills. Rather than setting small tasks that can be solved in a single session, the sessions are used to give students the opportunity to discuss coursework assignments with the lecturer on a regular basis and to receive feedback on their progress.

The students are provided with a web-based collaborative environment, which they are encouraged to use to share ideas and useful resources with the rest of the class.

Assessment  Assessment is by examination and coursework. Coursework accounts for 20% of the final mark and the end-of-year examination accounts for the remaining 80%.
3.3. COURSE DETAILS

Typically, the students will be required to complete four practical coursework assignments of varying size and complexity each year. The role of the first assignment is primarily to familiarize students with a distributed object-oriented technology. The second and most substantial assignment requires students to design, construct, evaluate and describe a substantial distributed application. The third assignment presents the students with a previously unfamiliar technology and requires them to perform sufficient research and experimentation to produce a short report summarizing the technology and describing how it can be used. In the final coursework assignment, students are required to compare the performance of a set of algorithms or policies used by operating systems by developing a software simulation.

In the final examination, students are required to answer five questions out of six and one of the questions, which relates to the practical work during the year, is compulsory. Each question typically requires a range of abilities, from recall through to synthesis and evaluation.

Recommended Texts  Various texts are suggested to students as sources of additional, in-depth information, including:


Students are also encouraged to locate their own sources of additional information.

CHAPTER 3. SENIOR SOPHISTER YEAR

3.3.11 Computer Vision and Robotics (4D4 Section B)

Lecturer: Dr Fergal Shevlin

Course Syllabus: This intended to give an broad overview of the course. It may change in detail if the lecturer thinks it appropriate to do so. The College Calendar is the definitive source of information.

- Introduction to image processing, computer vision, and their applications.
  (1 week approx.)

- Optics, illumination, and image sensors; image acquisition and representation.
  (1 week approx.)

- Fundamentals of digital image processing, local and neighbourhood operations, etc.
  (1 week approx.)

- Segmentation: region and boundary approaches.
  (1 week approx.)

- Image analysis: Hough transform, model fitting, etc.
  (2 weeks approx.)

- Robotics: Manipulator geometry, forward and inverse kinematics, parameterisations for representation and implementation of non-trivial spatial trajectories.
  (3 weeks approx.)

Assessment: Exam Worth 80% of final result. Projects Worth 20% of final result, one project for Section A, and one for Section B.
Chapter 4

Further Information


4.2 Freshman and Junior Sophister Engineering 2005/06

4.2.1 Overall Grading

Candidates are given an overall grade for the year at the annual examinations. This grade is based on the overall average percentage achieved, providing not more than two subjects are classed no lower than at least 35% but less than 40% or provided one subject is classed no lower than at least 30% but less than 35%. Individual subject results are published in percentage format ("f" denotes a percentage mark of less than 25% in any one subject). Full percentage results are available from the School of Engineering website: http://www.tcd.ie/Engineering/Courses/BAI/Results/ The full set of overall grades is set out below:
<table>
<thead>
<tr>
<th>Description</th>
<th>Grade</th>
<th>Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Class Honors</td>
<td>I</td>
<td>70% and above</td>
</tr>
<tr>
<td>Second Class Honors, First Division</td>
<td>II.1</td>
<td>60 - 69%</td>
</tr>
<tr>
<td>Second Class Honors, Second Division</td>
<td>II.2</td>
<td>50 - 59%</td>
</tr>
<tr>
<td>Third Class Honors</td>
<td>III</td>
<td>40 - 49%</td>
</tr>
<tr>
<td>Fail</td>
<td>F</td>
<td>An average of less than 40% or where failure in one or more subjects cannot be compensated as defined below</td>
</tr>
<tr>
<td>Exclude</td>
<td>EX</td>
<td>Examiners consider that the candidate has not made a serious attempt at the examinations</td>
</tr>
<tr>
<td>Result Not Available</td>
<td>NA</td>
<td>Candidate was absent with permission from the Senior Lecturer (due to medical or other grounds) and the result is incomplete</td>
</tr>
<tr>
<td>Result Withheld</td>
<td>RW</td>
<td>It may be necessary for academic or administrative reasons to withhold a result</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>WD</td>
<td>Candidate has withdrawn from the course</td>
</tr>
<tr>
<td>Repeat year</td>
<td>R</td>
<td>Candidates who are taking their year for the first time may repeat if they have an acceptable minimum standard (applies only at supplemental examinations)</td>
</tr>
<tr>
<td>Pass</td>
<td>P</td>
<td>Candidate has passed all subjects (applies only at supplemental examinations)</td>
</tr>
</tbody>
</table>

There is no formal distinction between direct achievement of a grade and through the application of the compensation procedure. After the examiners meeting, annual and supplemental examination results are published anonymously by student number in numerical order. Students who have failed the annual examination are required to take a supplemental examination in those subjects in which they have not satisfied the examiners. In the supplemental examinations there are no honor grades.

**Compensation**

Compensation is permitted in the annual examinations where:

- no more than two subjects are graded no lower than at least 35% but
less than 40% and each of the remaining subjects is passed and the overall average percentage is not less than 40%.

- where one subject is graded no lower than at least 30% but less than 35% and each of the remaining subjects is passed and the overall average percentage is not less than 40%.

Compensation is permitted in the supplemental examinations where:

- no more than one subject is graded no lower than at least 35% but less than 40% and each of the remaining subjects is passed and the overall percentage is not less than 40%.

4.2.2 Subject Results

Annual and supplemental examinations:
- a = absent with permission
- A = absent without permission or explanation automatic exclusion
- mc = medical certificate supplied to and accepted by the Senior Lecturer
- cr = credit for subject. Candidate exempt, e.g. on the basis of his or her performance in the scholarship examination
- gw = grade withheld

Supplemental examinations only:
- p = credit for subject where passed on previous occasion.

4.3 Senior Sophister Engineering 2005/06

4.3.1 Classes of degree

An honors BAI is awarded to candidates who reach the required total marks for the appropriate grade i.e. 420 (I), 360 (II.1), 300 (II.2), 240 (III), provided that not more than one individual mark falls below 40%. All other candidates who obtain an overall percentage mark of 40% or higher qualify for the Pass
(P) degree. Candidates not achieving an overall mark of 40% fail (F) and must present themselves at the next regular examinations in all subjects, including the project which must be re-submitted. Honors are only awarded at the first sitting of the degree examinations.

4.3.2 Contribution from Junior Sophister year

Except by special recommendation of the examiners, honors are awarded on either (a) the results of the annual B.A.I. examination of a students Senior Sophister year, or alternatively, on (b) the results of a students annual Junior Sophister examinations and subsequent annual B.A.I. examinations, taken together with the Junior Sophister result contributing 20 per cent to the combined mark.

Students in the following streams are assessed in accordance with (a) above:
- Computer Engineering
- Electronic Engineering
- Electronic/Computer Engineering (combined programme)

Students in the following streams are assessed in accordance with (b) above:
- Civil, Structural and Environmental Engineering
- Mechanical and Manufacturing Engineering

4.3.3 Marking of Projects

Each project should be marked by at least two examiners and the mean mark submitted. The Grade of project may be taken into account when discussing candidates marginally above or below the divisions between classes of degrees.

4.3.4 Marking Scheme

The marks assigned to the [six] component parts of the B.A.I examination are 100 in each of the subject options, 80 for Management for Engineers (4E1) and 120 for the project (4E2); 40% corresponds to 32/80 and 48/120, respectively.
4.4. HELP WITH ACADEMIC OR PERSONAL DIFFICULTIES

4.3.5 Publication of Results
Senior Sophister evaluation results will be published on Monday, 26 June, 2006.; Junior Sophister results will be published on Monday, 3 July, 2006.

4.4 Help with Academic or Personal Difficulties

ACADEMIC PROBLEMS: SOURCES OF ASSISTANCE

- other students in the class;
- the course lecturer;
- BAI Engineering class representatives;
- your personal tutor (or any other tutor if you cannot find yours), or the Senior Tutor;
- Head of Department or Dean of Engineering, Dr Brian Foley 608 1594 e-mail: brian.foley@tcd.ie;
- Students Union Education Officer, Ms Heledd Fychan (646 8439) e-mail: education@tcdsu.org

PERSONAL PROBLEMS: SOURCES OF ASSISTANCE

- Your personal tutor (or any other tutor if you cannot find yours), or the Senior Tutor

- Student Counsellors, 199/200 Pearse Street, College, tel: (01) 608 1407 or Niteline (9pm-2.30am) at 1800 793 793—Ms Deirdre Flynn, Ms Clare Maloney, Ms Vicky Panoutsakopoulou or Ms Tenia Kalliontzi

- Student Health Service, House 47:
  - Medical Director Dr David Thomas 608 1556
  - Medical Officer Dr Niamh Murphy 608 1556
  - Psychiatrist Dr Sinead O’Brien 608 1591
  - Psychiatrist Dr Maeve Daly 608 1591
  - Physiotherapist Ms Karita Cullen 608 1591
• Welfare Officer, Students Union: Mr Michael Miley, House 6, College (646 8437), email: welfare@tcdu.org

• Chaplains, House 27, College:

  Paddy Gleeson (Roman Catholic) 608 1260
  Alan McCormack (Church of Ireland) 608 1402
  Katherine Meyer (Presbyterian) 608 1901
  Richard Sheehy (Roman Catholic) 608 1260

• Contact persons if you feel you are being sexually harassed or the victim of bullying:

  Ms Anne-Marie Difflcy 608 2320;  Ms Sheila Maher 608 1573;
  Ms Ruth Torode 608 1025;  Ms Ann Mulligan 608 1239;
  Mr Pat Holahan 608 1091;  Dr Myra O’Regan 608 1834;
  Dr Tim Jackson 608 1501;  Ms Geraldine Ryan 608 1658.

• Any student, member of staff or other person with whom you feel able to discuss your problems.

NOTE: IF YOU HAVE A PROBLEM OF ANY SORT, PLEASE TALK TO SOMEONE STRAIGHT AWAY

TUTORS

The tutors responsible for engineering students are:
### 4.4. HELP WITH ACADEMIC OR PERSONAL DIFFICULTIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Code</th>
<th>College Address</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr MH Brady</td>
<td>Q8</td>
<td>Room G41, O’Reilly Institute</td>
<td>1786</td>
</tr>
<tr>
<td>Dr A Butterfield</td>
<td>M1</td>
<td>Room F13, O’Reilly Institute</td>
<td>2517</td>
</tr>
<tr>
<td>Mr WJ Dowling</td>
<td>A3</td>
<td>Electronic and Electrical Engineering</td>
<td>1741</td>
</tr>
<tr>
<td>Mr. D Geraghty</td>
<td>AC</td>
<td>Mechanical Eng., Parsons Building</td>
<td>1024</td>
</tr>
<tr>
<td>Mr LW Gill</td>
<td>V8</td>
<td>Civil Engineering, Museum Building</td>
<td>1047</td>
</tr>
<tr>
<td>Ms M Huggard</td>
<td>W3</td>
<td>Room 4.13, Oriel House</td>
<td>3690</td>
</tr>
<tr>
<td>Dr A Hughes</td>
<td>A7</td>
<td>Room G40, O’Reilly Institute</td>
<td>2459</td>
</tr>
<tr>
<td>Dr J Jones</td>
<td>98</td>
<td>F.12 O’Reilly Institute</td>
<td>1112</td>
</tr>
<tr>
<td>Mr CG Lyons</td>
<td>N1</td>
<td>Room 24, Parsons Building</td>
<td>1464</td>
</tr>
<tr>
<td>Mr BDR Misstear</td>
<td>T0</td>
<td>Civil Engineering, Museum Building</td>
<td>2212</td>
</tr>
<tr>
<td>Dr K Mosurski</td>
<td>K8</td>
<td>Room C31, Statistics Building</td>
<td>1830</td>
</tr>
<tr>
<td>Dr Brendan Murphy</td>
<td>A6</td>
<td>Room 131, Lloyd Institute</td>
<td>1830</td>
</tr>
<tr>
<td>Dr AJ O’Connor</td>
<td>W5</td>
<td>Civil Engineering, Museum Building</td>
<td>1822</td>
</tr>
<tr>
<td>D Donovan</td>
<td>G1</td>
<td>Mathematics, Hamilton Building</td>
<td>1698</td>
</tr>
<tr>
<td>Dr DW O’Dwyer</td>
<td>R3</td>
<td>Civil Engineering, Museum Building</td>
<td>2532</td>
</tr>
<tr>
<td>Dr K O’Kelly</td>
<td>S5</td>
<td>Mechanical Eng., Parsons Building</td>
<td>1367</td>
</tr>
<tr>
<td>Dr M O’Regan</td>
<td>G0</td>
<td>Room C25, Statistics Building</td>
<td>1834</td>
</tr>
<tr>
<td>Dr TLL Orr</td>
<td>S3</td>
<td>Civil Engineering, Museum Building</td>
<td>1204</td>
</tr>
<tr>
<td>Dr A Quinn</td>
<td>S6</td>
<td>Electronic and Electrical Engineering</td>
<td>1863</td>
</tr>
<tr>
<td>Ms M Sharp</td>
<td>S7</td>
<td>Room G36, O’Reilly Institute</td>
<td>2732</td>
</tr>
<tr>
<td>Dr Ciaran Simms</td>
<td>AD</td>
<td>Mechanical Eng., Parsons Building</td>
<td>3768</td>
</tr>
<tr>
<td>Dr M Stuart</td>
<td>N0</td>
<td>Room C34, Statistics Building</td>
<td>1014</td>
</tr>
<tr>
<td>Dr CD Walsh</td>
<td>W2</td>
<td>Room C14, Statistics Building</td>
<td>1731</td>
</tr>
<tr>
<td>S. Wilson</td>
<td>W2</td>
<td>Room 141, Lloyd Institute</td>
<td>1759</td>
</tr>
<tr>
<td>Senior Tutor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr C Laudet</td>
<td>L9</td>
<td>Senior Tutor’s Office, House 27</td>
<td>2004</td>
</tr>
</tbody>
</table>

Note: The reference number shown in the second column above is used during the publication of examination results.