UNIVERSITY OF WALES, ABERYSTWYTH

Department of Computer Science

Honours Modules
Level 1
1995-96

Contents

Level 1 Modules for Students Intending Honours in the Department of Computer Science

Introduction

The Level 1 modules in Computer Science intended for students wishing to proceed to single or joint honours degrees in Computer Science or Software Engineering, or to a degree in Information Science are described below. Service modules in Computing, i.e. modules intended for students taking honours in other subjects, are described elsewhere.

The Modules

The following Level 1 modules are available to honours students:

- **CS10010** Introduction to Computing (if no background in Computing);
- **CS11010** Software Development I;
- **CS11110** Software Development II;
- **CS12010** Software Laboratory I;
- **CS12110** Software Laboratory II;
- **MA12010** Basic Mathematical Techniques (for those with A level Maths or equivalent);
- **MA12510** Introductory Mathematical Techniques (for those without A level Maths or equivalent);
- **MA13310** Logic and Machines;
- **CS15010** Introduction to Computer Hardware;
- **CS15110** Computer Architecture.

All students wishing to proceed to a single or joint honours degree in Computer Science or Software Engineering must take the Computer Science modules, **CS11010**, **CS11110**, **CS12010** and **CS12110**, in their first year. All students who have no previous experience in computing must also take the introductory module, **CS10010**. Unless students are taking the Mathematics courses **MA10110** and **MA10210**, they must also take one of the Mathematical Techniques courses, **MA12010** or **MA12510** depending on A level experience, together with the Logic and Machines course **MA13310**. They may, if they wish, take the Computer Hardware modules, **CS15010** and **CS15110**, provided that the required prerequisites are met.

Note that **CS15010** must be be taken by all single and joint honours Computer Science students and by all engineering degree students *at some stage* in their degree.

Students admitted to study for a degree in Information Science must take **CS11010**, **CS11110**, **CS12010** and **CS12110**. Students should consult the departmental advisors for full details.

Students who wish to study some Level 1 Computing and who do not fall into one of the above categories are usually advised to take one or more of the service courses. These courses do not lead to modules in Part 2 Computer Science. Any such students who wish to take the Computer Science honours modules should consult the department at registration.

General Information and Advice

Assessment

Students on each of the modules described in this booklet are assessed during the semester in which the module is given. The grades achieved are published. In order to proceed to **CS11110** and **CS12110**, students must have followed **CS11010** and **CS12010**.

Details of the basis of the assessment are given under the individual course descriptions.

Attendance Requirements

Students are required to attend lectures; much of the material taught is not readily available in text books. They are also required to attend tutorials, timetabled practical sessions and other classes as may be arranged in connection with individual courses (see the individual course descriptions). Tutorials and practical sessions will be arranged at the start of the course, each in such a way as to take into account students' other academic commitments. Students who are unhappy with the groups to which they are assigned should see Mr. Graham Parker.

Illness or family bereavement are acceptable reasons for failure to attend lectures, tutorials or practical classes, but the department should be informed as soon as possible. Every effort should be made to inform the department *in advance* if you are unable to attend *for whatever reason*. A medical note is required if absence through illness lasts for more than one week; it should be submitted to the Student Health Centre but it is helpful if you can also inform the Departmental Office.

Disciplinary action may be taken against students who are persistently absent from lectures, tutorials, practical classes or examples classes, without good reason or who fail to submit course work punctually.

Commitment

The Tutorials of **CS11010** and **CS11110** are used to provide small group teaching (usually in groups of ten to twelve) to reinforce the material covered in lectures and to handle individual problems, including difficulties with assignments. In both **CS12010** and **CS12110**, there is a two hour supervised practical session per week; the purpose of this is to introduce students to the equipment and the computing environment and to assist with any problems that may arise. Throughout the course, worksheets intended for the practicals will be set at a rate of about one a week; students are expected to prepare for these in advance of the timetabled periods.

An advisory service is available both in the Sunlounge(C56) and the Solarium (B24) of the Llandinam building each day during term time. This is intended to provide emergency help with small difficulties; more substantial problems should be taken to the practical sessions.

Progress

The assessment methods for all modules are described under the detailed module descriptions in this brochure. Requirements in each degree scheme for progression to Part II are described in the College booklet "Degree Scheme Framework - Level 2 and 3".

A departmental booklet "Software Engineering Degrees - progress and assessment" explains particular features of the BEng and MEng schemes and the procedure for transfer between schemes.

Subject to the general regulations of the University, students who fail to meet the required standard in any of their modules may be eligible to resit that assessment during the summer or in the following year. Students who fail the Software Laboratory modules, **CS12010** or **CS12110** and are allowed a resit will be required to submit a resit project; the cost of returning to College in order to carry out the resit project is the student's responsibility.

A comment on practical work

The practical work is very important; one cannot pursue computing without becoming proficient at computer programming. What is more, it is time consuming and cannot be done in a hurry, so it is most important for you to keep abreast of practical assignments as they are set.

The following recommendations should be carefully noted:

- Spread out the work load for practical assignments evenly throughout the time allotted. Leaving work to accumulate near a deadline usually means that the computing facilities will be very busy and there will be long delays before your output gets printed – besides, computers have a nasty habit of breaking down just before a project deadline. (This is as true in industry as it is in university.)
- Plan what you intend to do in a session at a workstation, before you start.
- At busy times, you will make yourself very unpopular if you try to hog a workstation for long periods.

One final piece of advice.

Although zeal in one's studies is generally accounted a very good thing, there are dangers. It is very easy for some people to become hooked on computing, to the detriment of their studies in other subjects. If you find that you have a talent for computing and that you enjoy it, beware of spending a disproportionate part of your time doing it. We look forward to welcoming you into Part 2; but this can only happen if you pass Part 1 in your other courses as well. If you find that you can complete the programming assignments well within the set time, spend the rest of the time working on your other
1.4 courses.

Descriptions of Modules

The rest of this brochure provides details for each module.

CS10010 Introduction to Computing

General Description

This is a basic introduction to the use of computers, intended for students with no previous experience of computers. It is available to students in any faculty. Students who complete the course successfully will be able to use personal computers for word processing and will be familiar with files, directories, discs and printers. They will also be able to use the more common utility programs (e.g. the file manager) and have some familiarity with the ideas of spreadsheets and databases.

Commitment	Number of lectures	12
	Number of workshops	-
	Number of practicals	20 x 2 hours
Co-ordinator	Dr. John Hunt	
Other staff	Not yet known	
Pre-requisites	None	
Co-requisites	None	
Incompatibilities	Not available to students who have significant experience of using computers	
Assessment	Assessed coursework - 100%	
Timing	This module is offered in both Semester 1 and Semester 2	

Aims

This module provides an introduction to the use of computers by:

- introducing the range of computer equipment, its availability and capability and thereby encouraging an understanding of the principles *and*
- by developing skills which will both be of value outside the course and will give experience and insight to strengthen the broader understanding.

Consideration of the applicability of the technology to students' main disciplines will be encouraged. This module is not designed to develop an understanding of technical aspects of computing.

Objectives

On successful completion of the module, students should have an understanding of:

- what hardware components are found in typical computer systems and broadly what their functions are
- typical cost and performance characteristics of the range of technologies and products such as: processors, primary store, secondary store, displays, printers, pointing devices, and digitisers
- the range of software (from operation systems, file systems and display managers, through utility programs, language systems and general packages to highly specialised packages) and an outline of its value to the user
- the potential benefits of computer technology to academics, in particular in the areas of document production and of data storage and retrieval

- the costs and possible disadvantages of using the technology
- legal and ethical problems in using the technology.

On successful completion of the module students should be able to:

- create and manage disc file systems under Microsoft Windows
- produce effective and well designed documents of significant complexity using a word-processor of reasonable sophistication
- create graphics (using bit-mapped and object based packages) and charts and include them in wordprocessed documents
- create and use simple spreadsheets
- create and use simple, single table databases
- use electronic mail
- use on-line services such as the CD-ROMS and OPACs
- evaluate and select appropriate equipment for a range of non-technical applications; this to include software and display, printing and storage devices

Syllabus

1. INTRODUCTION — 1 lecture
   Course organisation. Aims and Objectives.
2. SURVEY OF SOFTWARE — 2 lectures
   Operatings systems; user interfaces, utilities; applications programs; user written programs.
3. SURVEY OF HARDWARE — 2 lectures
4. GRAPHICS APPLICATIONS — 1 lecture
5. DATABASES — 2 lectures
6. SPREADSHEETS — 1 lecture
7. DATA PROTECTION LEGISLATION — 1 lecture
8. DOCUMENT PREPARATION — 1 lecture
9. ADVANCED APPLICATIONS — 1 lecture
10. MICROSOFT WINDOWS — 1 practical
    General use and manipulaton of Windows.
11. E-MAIL; FILES I. — 1 practical
    Creation, manipulation and maintenance of a directory structure.

12. WORD PROCESSOR — 5 practicals
    General use of Microsoft WORD.
13. FILES II — 2 practicals
    Revision of the use of directory structures following practical experience of them.
14. GRAPHICS — 2 practicals
    Bit mapped and object orientated graphics applications (Mircrosoft Paint and Draw).
15. SPREADSHEETS — 2 practicals
    A brief introduction to the potential of spreadsheets (using Microsoft Excel).
16. DATABASES — 2 practicals
    Experience of setting up a simple database; search and report generation.
17. ASSIGNMENTS — 4 practicals
    Time is allocated for the work on assignments.

Booklist

The following should be consulted for different approaches or for further information:

Nancy B. Stern and Robert A. Stern. *Computing in the Information Age*. John Wiley and Sons Inc, New York, 1993. 1.2

CS11010 Software Development I

General Description

This module introduces students to the basic concepts of programming in the context of a professional approach to software development. Emphasis is placed on the design of software using hierarchical decomposition and its subsequent implementation using the Ada programming language. In addition to the technical material, the development of personal communication skills is an important aspect of this course.

The module is intended primarily for students wishing to read for honours in the Computer Science Department.

Commitment	Number of lectures	24
	Number of workshops	12 (first year tutorials)
	Number of practicals	An activity weekend
Co-ordinator	Dr. Mark Ratcliffe	
Other staff	Not yet known	
Pre-requisites	None	
Co-requisites	**CS12010**; also **CS10010** for students with no previous experience of computers.	
Incompatibilities	None	
Assessment	Written exam - 100%	
Timing	This module is offered only in Semester 1	

Aims

This module aims to introduce students to the idea of problem solving and algorithm design together with subsequent implementation using the programming language Ada. Emphasis is placed on distinguishing between the design of a software product and its implementation.

Objectives

The course concentrates on a professional approach to software development. On successful completion of the module, students should:

- have a good understanding of the concepts of problem abstraction and program design
- be capable of realising their design in the Ada programming language
- have an understanding of the UNIX operating system at a level to support their programming requirements.

This module is very tightly coupled with **CS12010**, consequently the syllabus shown here covers material lectured in both **CS11010** and **CS12010**.

Syllabus

1. WELCOME AND PREVIEW — 1 lecture
   Introduction to the department and the course.
2. MANAGEMENT ISSUES AND PROFESSIONAL CONDUCT — 2 lectures
   The growing role of computing, software, the need for software engineering and professionalism. Software engineering: requirements, design, implementation, testing, maintenance, quality, need for documentation.
3. SOFTWARE CRISIS, ABSTRACTION, ALGORITHMS AND PROGRAMS — 1 lecture
4. THE DESIGN OF ALGORITHMS AND INTRODUCTION TO ADA — 2 lectures
5. THE REPRESENTATION OF ALGORITHMS — 2 lectures
6. PROGRAMMING CONSTRUCTS, EXPRESSIONS, SIMPLE TYPES — 3 lectures
7. TESTING - TECHNIQUES AND AIDS FOR ERROR DETECTION — 2 lectures
8. INFORMATION HIDING - INTRODUCTION TO PROCEDURES — 4 lectures
9. ABSTRACTION SUPPORT USING PACKAGES — 5 lectures
10. COMPOSITE TYPES - ARRAYS AND RECORDS — 5 lectures
11. INTRODUCTION TO TELEMATICS — 2 lectures
    Telematic applications - mailers, teleworking, the internet.

Booklist

It is considered essential to purchase the following:

J G P Barnes. *Programming in Ada*. Addison-Wesley, Wokingham, England, 4th. edition, 1993.

J.E.Cooling, N.Cooling, and J.Cooling. *Introduction to Ada*. Chapman and Hall, 1993.

N.B. Only one of the above need be purchased. Cooling *et. al.* is better suited for those who have no programming experience.

The following should be consulted for different approaches or for further information:

J. Glenn Brookshear. *Computer Science - An Overview*. Benjamin/Cummings, Redwood City, CA, 3rd. edition, 1991.

N.B. A good text for students who have no previous computing.

4.2

CS11110 Software Development II

General Description

This module is a continuation of **CS11010**. In addition to introducing students to more advanced aspects of software design and implementation, the module covers introductory material in telecommunications, databases and language processing.

The development of personal communication skills is an important aspect of this course.

Commitment	Number of lectures	24
	Number of workshops	12 (First year tutorials)
	Number of practicals	An activity weekend
Co-ordinator	Dr. Mark Ratcliffe	
Other staff	Not yet known	
Pre-requisites	**CS11010**, **CS12010**	
Co-requisites	**CS12110**	
Incompatibilities	None	
Assessment	Written exam - 100%	
Timing	This module is offered only in Semester 2	

Aims

This module builds on the material covered in **CS11010** to further the development of participants' programming skills by:

- introducing more advanced facilities that are available to the software engineer to improve the robustness, reusability and maintainability of software;
- applying these skills in practice using the Ada programming language.

Objectives

The course concentrates on instilling a professional approach to software development. Further programming concepts are introduced in the context of developing more robust and reliable software.

On successful completion of the module, students should:

- appreciate the need for a professional approach to software development;
- understand the importance of software design, coding and testing;
- be able to analyse a problem and produce high quality software designs;
- have a thorough understanding of Ada to the extent that they can take full advantage of all of the major facilities within the program language;
- produce high quality software that is robust, reliable, reusable and maintainable.

This module is very tightly coupled with **CS12110**, consequently the syllabus shown here is lectured in both **CS11110** and **CS12110**.

Syllabus

1. PERSONAL COMMUNICATIONS SKILLS I — 2 lectures
   CV development. Team roles; Belbin methodology and self assessment. Interpersonal skills - achieved in associated Team Skills weekends. (This subject is ongoing and is continued through the tutorial system)

2. MORE ADVANCED DESIGN AND IMPLEMENTATION OF COMPUTER PROGRAMS — 14 lectures
   Developing robust programs - illustrated by exception handling. Building abstract data types - case studies. Handling persistent data - advanced input/output and files.

3. INTRODUCTION TO TELECOMMUNICATIONS — 3 lectures
   Network protocols. Communication technologies. Standards.

4. LANGUAGE PROCESSING AND ASSEMBLY LANGUAGE — 3 lectures
   Compilation System Components; what software tools do you use to create, compile, bind and run an Ada program. Phases of a compiler, and what happens during each phase. Linkers and loaders. Alternative approaches to language processing: interpreters and assemblers.

5. SOFTWARE ENGINEERING; MANAGEMENT ISSUES AND PROFESSIONAL CONDUCT — 5 lectures
   The software life cycle and the various documents needed. Management of software projects. Planning, reviews. Software requirements; identification, specification, example of information specification, completeness, consistency, ambiguity. The legal framework. Professional conduct. The BCS code of conduct. The Data Protection Act. The Computer Misuse Act.

Booklist

It is considered essential to purchase the following:

J G P Barnes. *Programming in Ada*. Addison-Wesley, Wokingham, England, 4th. edition, 1993.

J.E.Cooling, N.Cooling, and J.Cooling. *Introduction to Ada*. Chapman and Hall, 1993.

N.B. Students do not need to obtain both these Ada texts. Choose the one whose style you like. Cooling *et. al.* is better suited for those who have no programming 4.2

CS12010 Software Laboratory I

General Description

This module consists of practical work and lectures designed to support **CS11010**; students will learn how to edit, compile, run and test simple programs in Ada.

Commitment	Number of lectures	12
	Number of workshops	12
	Number of practicals	12 x 2hours
Co-ordinator	Dr. Mark Ratcliffe	
Other staff	Not yet known	
Pre-requisites	None	
Co-requisites	**CS11010**	
Incompatibilities	None	
Assessment	Assessed coursework - 100%	
Timing	This module is offered only in Semester 1	

Aims

This module is very tightly coupled with **CS11010**, consequently the lecture slots available in this module present material described in the **CS11010** syllabus.

The intention of this module is to develop the programming skills of participants through supervised practical sessions.

Objectives

The practicals associated with this module provide participants with the opportunity to carry out a series of assignments associated with the theoretical material presented in lectures.

On successful completion of the module, students should:

- have a practical understanding of the Ada programming language
- be able to use the workstations to develop their programs to meet the specified requirements.

Most of the syllabus is shared with that listed under **CS11010**. Topics described here are specific to **CS12010**.

Syllabus

1. COMPUTER SYSTEM APPRECIATION — 2 lectures
   Introduces the student to the basic computer organisation. Emphasis is placed on the relationships between hardware, architecture and software.
   These presentations are illustrated by reference to the Computer systems at Aberystwyth.
2. USING THE COMPUTING FACILITIES AT ABERYSTWYTH — 3 lectures
   Introduction to UNIX and the Ada program development system.

Booklist

Not appropriate. 4.1

CS12110 Software Laboratory II

General Description

This module consists of practical work and lectures designed to support **CS11110**; students will develop more complicated programs in Ada and learn to use some of the more advanced tools available.

Commitment	Number of lectures	12
	Number of workshops	12
	Number of practicals	12 x 2 hours
Co-ordinator	Dr. Mark Ratcliffe	
Other staff	Not yet known	
Pre-requisites	**CS11010**, **CS12010**	
Co-requisites	**CS11110**	
Incompatibilities	None	
Assessment	Assessed coursework - 100%	
Timing	This module is offered only in Semester 2	

Aims

This module is very tightly coupled with **CS11110**, consequently the lecture slots available in this module present material described in the **CS11110** syllabus.

The intention of this module is to enhance the programming skills of participants through supervised practical sessions.

Objectives

On successful completion of the module, students should:

- have extensive practical experience in analysing a problem and producing high quality software designs;
- have a thorough understanding of Ada to the extent that they can take full advantage of all of the major facilities within the program language.

Though most of the syllabus is shared with that listed under **CS11110**, the following topics are specific to CS12110.

Syllabus

1. UNIX: AN ENVIRONMENT TO SUPPORT COMPONENT REUSE? — 2 lectures
   An examination of basic UNIX components e.g. sort, uniq and the pipe.
2. INTRODUCTION TO DATABASES — 2 lectures
   Information systems; examples, structure of an information system, choosing a data repository; manual, file based or database. Tools provided by a database management system; e.g. SQL*Plus, SQL*Forms, embedded SQL

3. APPLICATIONS OF COMPUTERS 3 lectures
   Lectures will be given by various members of staff and will describe a range of innovative applications of which they have experience.

4. FOR NON-**CS15110** STUDENTS
   The following areas will be covered in special workshops: Recap of some basic ideas from computer architecture; Examples of some simple instructions; Assembly language as a convenient way of programming machines at this level; What an assembler does; Use of addressing modes.

Booklist

Not appropriate. 4.2

CS15010 Introduction to Computer Hardware

General Description

There is a very close relationship between system software design, system hardware design and the performance of the system as a whole. It is advantageous for computer scientists and software engineers to have an empathy for this relationship and therefore this module concentrates on the principles involved rather than fine detail. The module begins with a study of simple logic functions and how they can be combined to give increasing complexity and functionality. From this the course builds to a study of the operation of a typical system as a whole and the interaction between the various components within such a system. The laboratory work associated with the course is designed to consolidate the lecture material and gives practical experience in the use of various components and associated test equipment.

Commitment	Number of lectures	24
	Number of workshops	0
	Number of practicals	10 x 2 hours
Co-ordinator	Dr. John Hunt	
Other staff	Mr. Graham Parker	
Pre-requisites	None	
Co-requisites	**CS11010**	
Incompatibilities	None	
Assessment	Assessed coursework - 50%	
	Written exam - 50%	
Timing	This module is offered only in Semester 1	

Aims

To provide an appreciation of what computer hardware is, what it does, how it does it and how it influences the execution and performance of software.

- Introduction to logic and its hardware implementations.
- An appreciation of logic families.
- Familiarisation with small embedded systems.
- Familiarisation with interfacing such systems with the real world.
- An appreciation of technological advances in the hardware domain.

Objectives

On successful completion of this module, students will have a broad functional understanding of computer hardware. In addition, students will have detailed knowledge of:

- the function and application of combinational and sequential logic;
- the appropriate use of differing I.C. outputs;
- fundamental system components, their interactions and their dependencies;

- various memory implementations together with their suitability for given applications;
- the function and use of various peripheral interface devices;
- microprocessor evolution, together with an appreciation of how the area may develop in the future.

Syllabus

1. NUMBER SYSTEMS AND LOGIC — 1 lecture
   Decimal; Binary; Octal; Hexadecimal; Boolean Logic.
2. COMBINATIONAL LOGIC — 2 lectures and 2 practicals
   Logic functions; Truth tables; Minterms; Boolean relationships; De Morgan's Theorem; Half Adders; Full Adders; Parallel Adders.
3. SEQUENTIAL LOGIC — 2 lectures and 2 practicals
   RS Latch; Clocked devices; D-Type Latch; JK Flip-Flop; Storage registers; Counters; Shift registers; Logic representations.
4. INTEGRATED CIRCUITS - CHIPS — 2 lectures
   Packaging; Logic levels; Logic families; Totem Pole outputs; Open Collector outputs; Tri-State outputs; Comparators.
5. BASIC COMPUTER ARCHITECTURE — 6 lectures and 6 practicals
   System components: Processor, RAM, ROM, Input/Output; Buses; Memory addressing; Memory maps; Line Decoders; I/O Mapping; Execution cycles; Bus timing; The Synchronous Bus; The Asynchronous Bus.
6. INTERRUPTS — 1 lecture
   Application; Service routines; Priorities; Vectors; Power-On Reset.
7. MEMORY TYPES — 3 lectures
   Static RAM; Dynamic RAM; DRAM Controllers; Masked ROM; EPROM; EEPROM; Data Widths; PROM Decoding.
8. PERIPHERAL DEVICES AND INTERFACING — 4 lectures
   Handshaking; Parallel Interfaces; Serial Interfaces; Timers; Watch Dog Timers; Direct Memory Access; Backplane Buses.
9. MICRO-PROCESSOR EVOLUTION — 3 lectures
   Micro-Processor Families; Motorola 68000 series; Intel 80X86 series; CISC and RISC; The Transputer; Single-Chip Processors; Parallelism; Branch Prediction; Pipelining; The Pentium; The PowerPC; The future?

Booklist

It is considered essential to purchase the following:

Alan Clements. *Principles of Computer Hardware*. Oxford, 1991.

Ian Sayers et. al. *Principles of Microprocessors*. CRC Press, 1991.

N.B. Only one of these books will be required by each student.

The following should be consulted for different approaches or for further information:

Alan Clements. *Microprocessor Systems Design*. PWS-Kent, 1992.

John Fulcher. *Introduction to Microcomputer Systems: Architecture and Interfacing*. Addison-Wesley, 1991.

V.C. Hamacher et, al. *Computer Organisation*. McGraw-Hill, 1989.

M. Rafiquzzaman et. al. *Modern Computer Architecture*. West, 1989.

2.7 A.J. van de Goor. *Computer Architecture and Design*. Addison-Wesley, 1989.

CS15110 Computer Architecture

General Description

This module builds on the material covered in **CS15010** providing a detailed study of the typical units constituting a microprocessor system. Emphasis is placed on Low-level language programming and associated software engineering issues together with microprocessor architecture and interfacing to the analogue and digital world. Semi-conductor technology is also covered at a materials level, together with a study of fabrication and design techniques. Practicals combine the assembly language programming aspects with the architecture and interfacing issues, to give a practical experience of the hardware/software interface.

Commitment	Number of lectures	24
	Number of workshops	-
	Number of practicals	10 x 2 hours
Co-ordinator	Dr. John Hunt	
Other staff	Mr. Graham Parker	
Pre-requisites	A grade C or better in GCSE Physics, Electronics, or Modular Science including modules on electricity *and* **CS15010**	
Co-requisites	None	
Incompatibilities	None	
Assessment	Assessed coursework - 50% Written exam - 50%	
Timing	This module is offered only in Semester 2	

Aims

The module aims to give an appreciation and experience of the interface between hardware and software along with the role of low-level languages. In addition, it is the intention that students will gain an understanding of the internal architecture and operation of microprocessor devices and the practicalities of interfacing to peripheral devices. An insight to Computer Aided Design environments will be provided by the practicals associated with full-custom design.

- To introduce assembly language programming.
- Raise awareness of the appropriate role of low-level languages.
- Highlight the intricacies and pitfalls of low-level language implementations.
- Application of established software engineering practices.
- To provide an understanding of the internal architecture of microprocessors in general.
- Give an awareness of the issues involved in interfacing to digital and analogue systems.

Objectives

It is not the objective of this module to qualify students as expert assembler language programmers or expert hardware engineers. Students will however, be expected to have a strong empathy toward the software/hardware /architecture interfaces, together with a sound practical experience of a range of typical devices and implementations. Specifically this will include:

- A comprehensive appreciation of the Motorola 68000 assembler language in terms of syntax, semantics and pitfalls.
- A detailed knowledge of a subset of the Motorola 68000 assembler language instruction set.
- An understanding of the importance of, and role of, a stack in the correct operation of a modular piece of code and as a parameter passing mechanism.
- An understanding of the use and application of bit-wise and byte-wise manipulation instructions.
- An understanding of the appropriate use of assembler language programming.
- A working knowledge of the internal registers of a typical microprocessor.
- A detailed knowledge of the Motorola 68230 PIT.
- The purpose and operation of Operational Amplifiers in the context of interface design.
- An understanding of ADC and DAC devices; Resolution; Linearity; Sampling rates; Aliasing; Resistor networks.
- An appreciation of the role and operation of Cache.
- A basic knowledge of semi-conductor fabrication.

Syllabus

1. INTRODUCTION — 1 lecture
   Course Organisation; Introduction to low-level languages; Characteristics of low-level languages; Processor characteristics.
2. MOTOROLA 68000 ASSEMBLER LANGUAGE — 9 lectures and 5 practicals
   Assembler language programmer's model - Register sets; Layout and format of code; Structure and design; Introduction to Instruction Set; Subroutines; Use of a stack; Role of Stack Pointer; Assembler directives; Control of thread of execution - Jumps and Branches; Role of the Status Register; Role of Programme Counter; Equivalent high-level language constructs; Bit manipulation; Addressing Modes; Applications; Subroutine workspace; Introduction to assembler environment; Use of environment and 68000 Tutor Boards.
3. PERIPHERALS AND INTERFACING — 6 lectures and 4 practicals
   68230 PIT: Detailed look at use of; Use of data sheets. Operational Amplifiers: Ideal characteristics; Virtual Earth; Buffers and Impedance matching; Comparators.
4. ARCHITECTURE — 3 lectures
   Internal Registers; Control Units: Hard-wired and Microcoded; Instruction types; Instruction cycles.
5. CACHE — 1 lecture
   Performance considerations; Fully associative cache; Direct mapped cache.
6. SEMI-CONDUCTOR TECHNOLOGY — 4 lectures and 1 practical
   Principles of CMOS technology; Materials; Doping; Fabrication of CMOS transistors; Fabrication of large scale devices; Design rule checking and relevance to computer scientists.

Booklist

It is considered essential to purchase the following:

Alan Clements. *Principles of Computer Hardware*. Oxford, 1991.

Ian Sayers et. al. *Principles of Microprocessors*. CRC Press, 1991.

N.B. Only one of these books will be required by each student.

The following should be consulted for different approaches or for further information:

Alan Clements. *Microprocessor Systems Design*. PWS-Kent, 1992.

Alan Clements. *Principles of Microprocessor Design*. PWS-Kent, 1992.

John Fulcher. *Introduction to Microcomputer Systems: Architecture and Interfacing*. Addison-Wesley, 1991.

King and Knight. *Programming the M68000*. Benjamin/Cummings, 1987.

Leventhal. *68000 Assembly Language Programming*. McGraw-Hill, 1986.

M68000 users manual. Motorola. 2.5