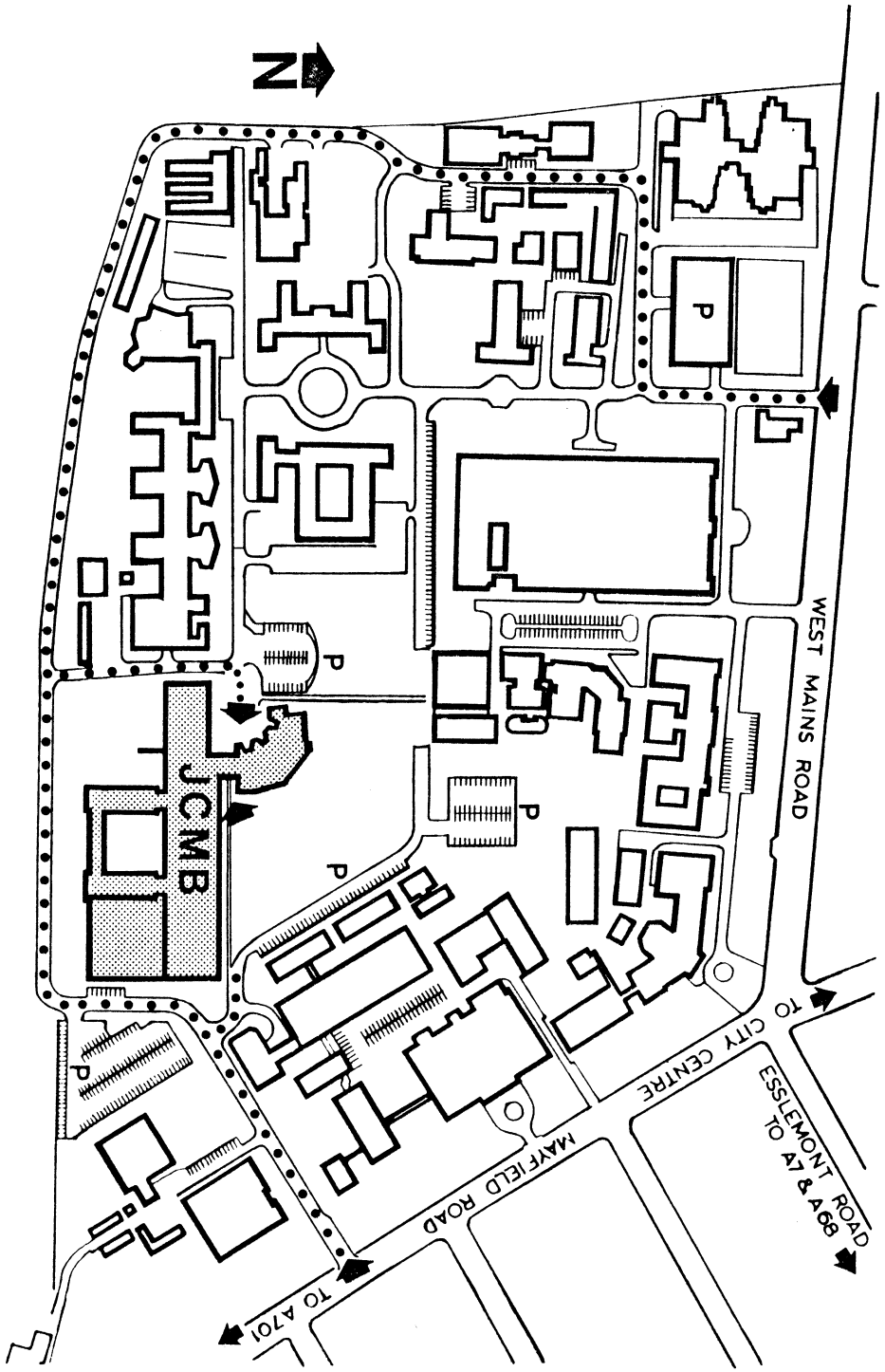
Department of Computer Science

Handbook
1977-78

University of Edinburgh



SITUATION OF THE JAMES CLERK MAXWELL BUILDING IN THE KING'S BUILDINGS SITE

Contents

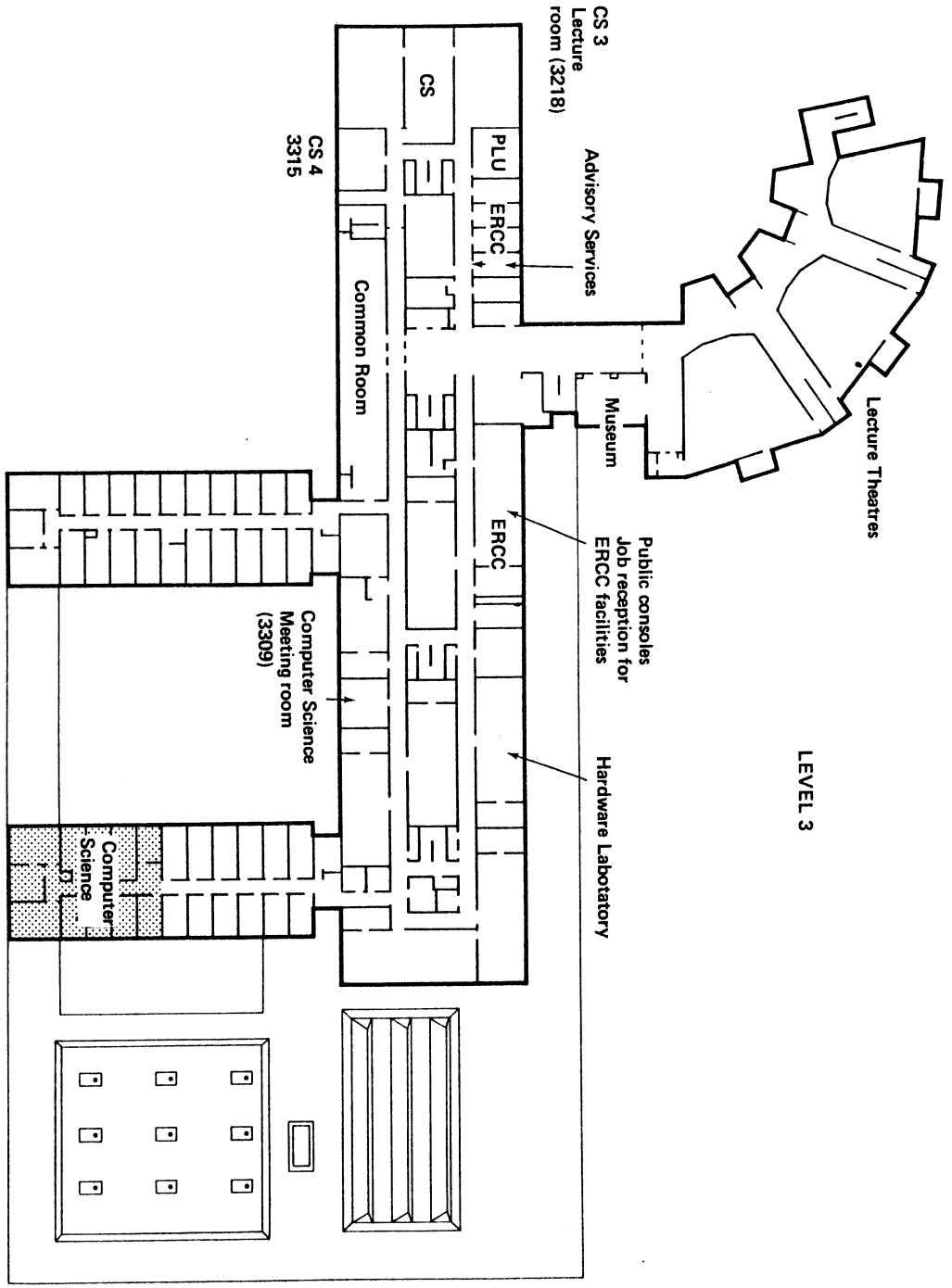
Editor, J.Tansley, Nov 1977.

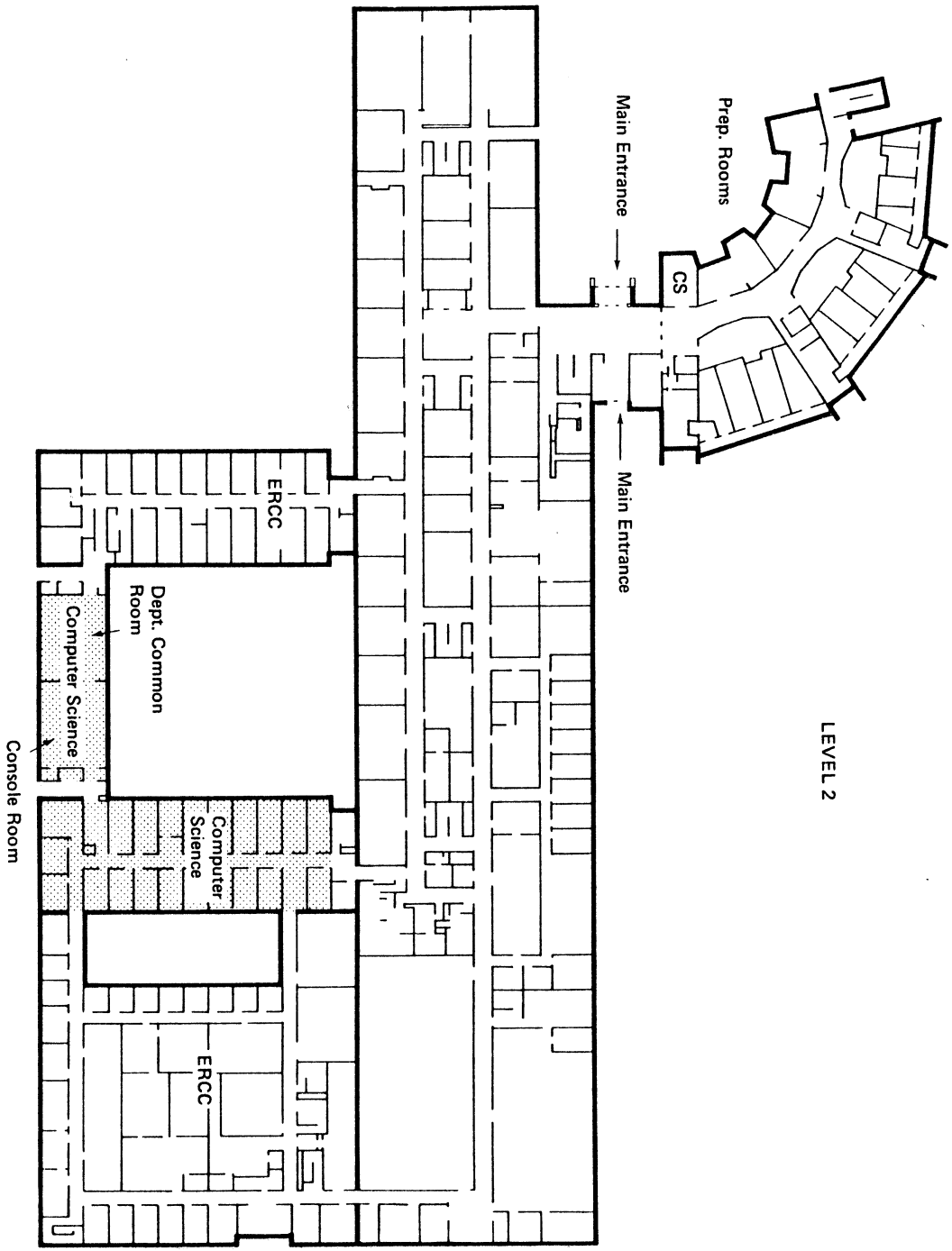
1.0 Introduction.

This handbook is intended as an informal summary of the organization and work of the Department of Computer Science at Edinburgh University. It is hoped that it will serve as useful introduction to the Department for visitors, new students and staff. In most instances fuller details of the items mentioned in this booklet may be obtained by consulting the appropriate documentation.

The Department of Computer Science was created in 1966 when the Computer Unit was subdivided as a consequence of the Flowers Report on computing for the Universities and Research Councils. The Department inherited responsibility for teaching and research in Computer Science. There already existed a Postgraduate Diploma and a first year course in the subject, together with a large research project, the Edinburgh Multi-Access Project, from which sprang the interactive computing system (EMAS) which provides the service now enjoyed by the University. As funds became available, computers were bought and the number of teaching staff was increased from the initial half-dozen. Second and third year courses were introduced in 1968 and 1970 - the latter made possible by the cooperation of the Mathematics Department in setting up a joint Honours School. At this time the Department moved into its new quarters in the JCMB, thereby acquiring the laboratory space that was needed for the sort of teaching that it regarded as central to courses in Computer Science. By 1972 it became possible to launch a Final Honours year in the subject and an Honours degree in Computer Science alone became available. Since then, teaching effort has been devoted to consolidation of the course, to upgrading the laboratory facilities, to improving its structure and its content and to ensuring that the improvements in our understanding of computation are reflected in what we teach. Research has not been static during this period. A large group grew in the Department, working on Computer Aided Design and graphics, and a large group is now growing in the field of the Theory of Programming. The project on multi-access systems came to an end and was largely replaced by work on software for mini-computers; a new multi-access project developed, has achieved its aims and is now coming to an end: work on the measurement and analysis of systems is now building up, linked with investigations of systems having new structures.

We feel that the Department at Edinburgh is well attuned to a subject as rapidly changing as computer science --- able to follow a natural pattern of research that is reflected in the areas we choose to investigate and teach.





2.0 Location

The Department of Computer Science is located in the James Clerk Maxwell Building (JCMB). The JCMB is some two miles from the central University area and is located on the Science Faculty Kings Buildings site. The accomodation of this large and dominant building is shared by six departments. Certain communal facilities are provided; these include:-

a common room on the third floor, where drinks and snacks can be purchased,

teaching and tutorial rooms,

three large lecture theatres,

a library, on the fourth floor, which contains a reasonable selection of books and journals relevant to computer science.

All the teaching for the second, third and fourth year degree courses in Computer Science takes place in the JCMB. First year teaching is conducted at the Appleton Tower, a building complex that is part of the University's central site. In the Appleton Tower the shared facilities include five large lecture theatres on level 3; while the Department has a lecture room, tutorial rooms, a work room and a console room containing 16 consoles and a line printer on-line to the University's main computing facility.

The Departmental accommodation in the JCMB consists of:-

staff rooms, (levels 1, 2 & 3 of block P),

a meeting room (3309),

a common room, (west end of block O/P), which contains reports distributed by other Computer Science Departments and some trade journals and newspapers.

a kitchen, (level 1 block O), where coffee, other hot drinks, snacks etc can be made at any time of the day,

student work rooms, (level 1, block O), adjacent to the kitchen area,

two large machine halls, (level 1), housing most of the the Department's equipment and an electronics workshop,

a hardware laboratory, (3304),

a communal console room, (east end of block O/P), containing 12 terminals on-line to the University Computers.

All of these facilities are on the first three levels of the JCMB and towards the east end of the building. Plans showing the relative position of these rooms and laboratories are given in the last section (13), of this document.

2.1 Access

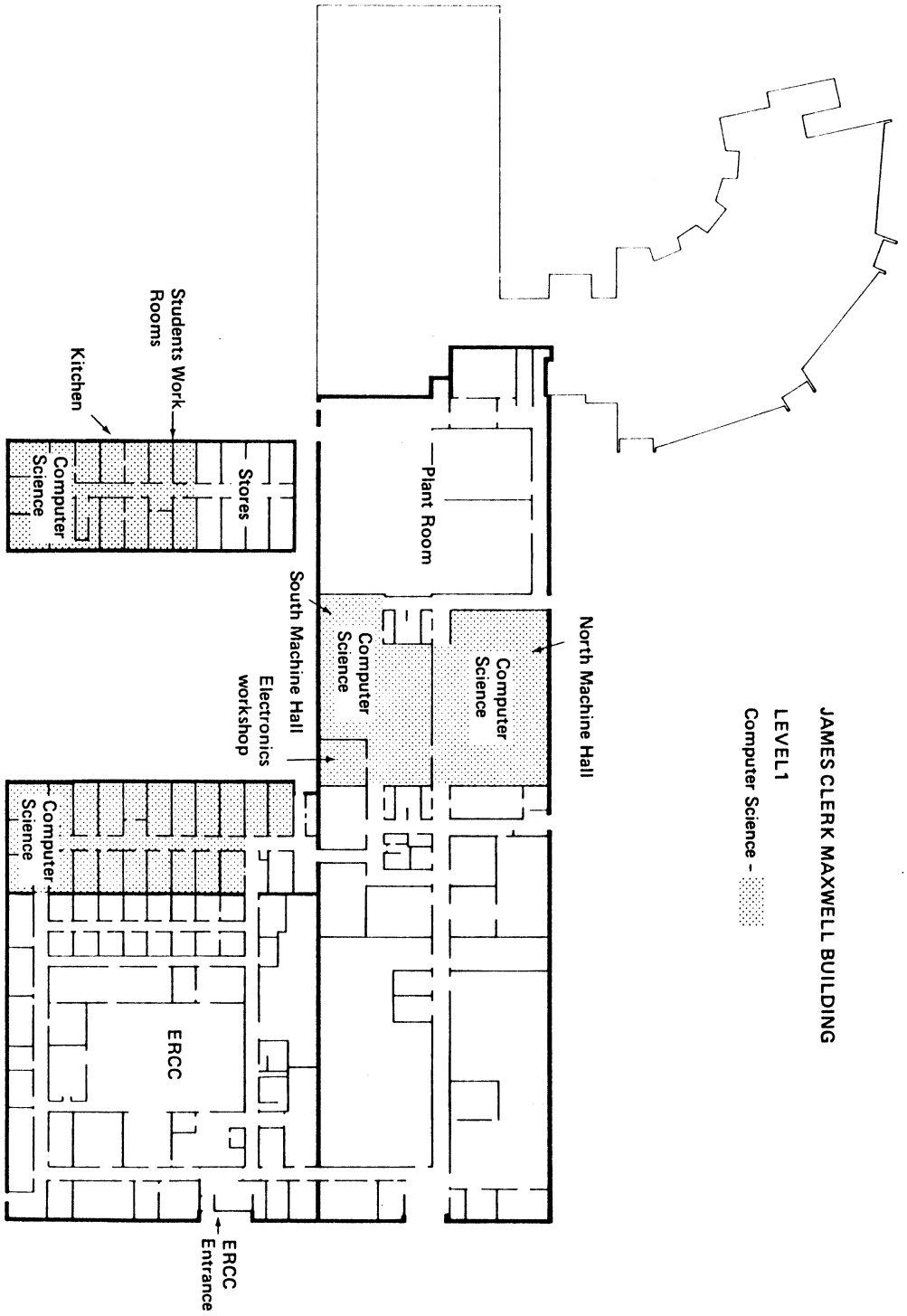
Staff, postgraduates and students in their honours years have access to the JCMB 24 hrs per day 7 days per week. Access to the library is possible outside normal hours by signing out a key, obtainable from the main servitors' lodge.

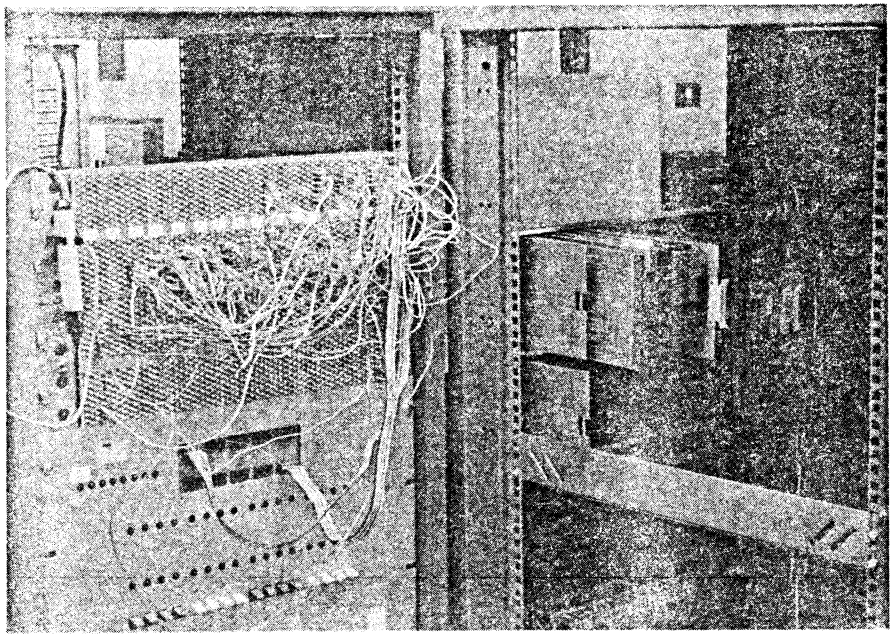
Other students have access to the JCMB from:-

8.00 am until 10.00pm week days
8.00 am until 1.00 pm Saturdays.

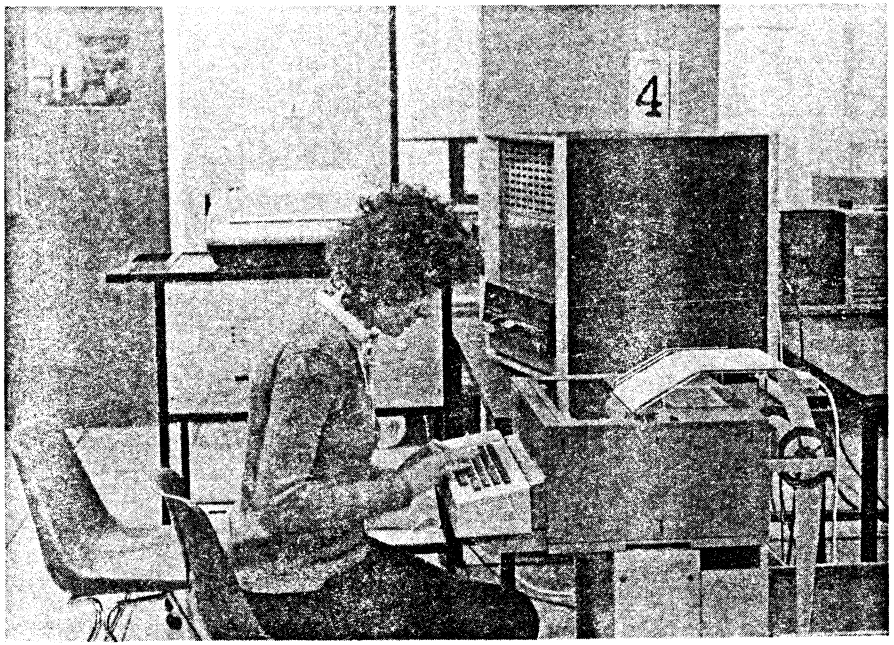
The ERCC entrance is open from:-

8.00 am until 6.30 pm week days only.





Register transfer kits, showing the construction methods.



Programming a satellite Interdata connected to ISYS.

3.0 General Departmental Organisation.

Many of the Department's activities are organised and discussed at regular meetings held on the first Monday of every month. These meetings are open to all members of the Department and everyone, including students, is encouraged to attend. Through these meetings detailed planning and day to day administration is delegated to a number of committees.

Technical Services Committee

Chairman:- P.D.Schofield, Secretary:- J.Tansley

This committee is responsible for the day to day running and maintenance of the Department's equipment and the design and construction of new equipment. It also plans the development of general teaching and research facilities in the Department.

Syllabus Committee

Chairman:- A.Wight.

Reviews and develops the syllabus for undergraduate courses.

Postgraduate Committee

Chairman:- D.J.Rees

Reviews admissions and progress of postgraduate students. Also responsible for research seminars and postgraduate courses.

Seminar Committee

Chairman:- J.C.Adams

Organises the weekly Departmental Seminar.

Examination Committee

Chairman:- P.D.Schofield

Assess the weighting and composition of the undergraduate examinations.

3.1 Representation on Committees External to the Department.

The following people also represent the Department's interests on the boards and committees listed below.

JCMB Library Committee.

N.Shelness.

JCMB Users Committee

P.D.Schofield, F.Stacey.

Sub-Committees of the JCMB Users Committee.

Accommodation.

F.Stacey.

General Services.

N.Shelness.

Maintenance and Technical Services.

T.Buckley.

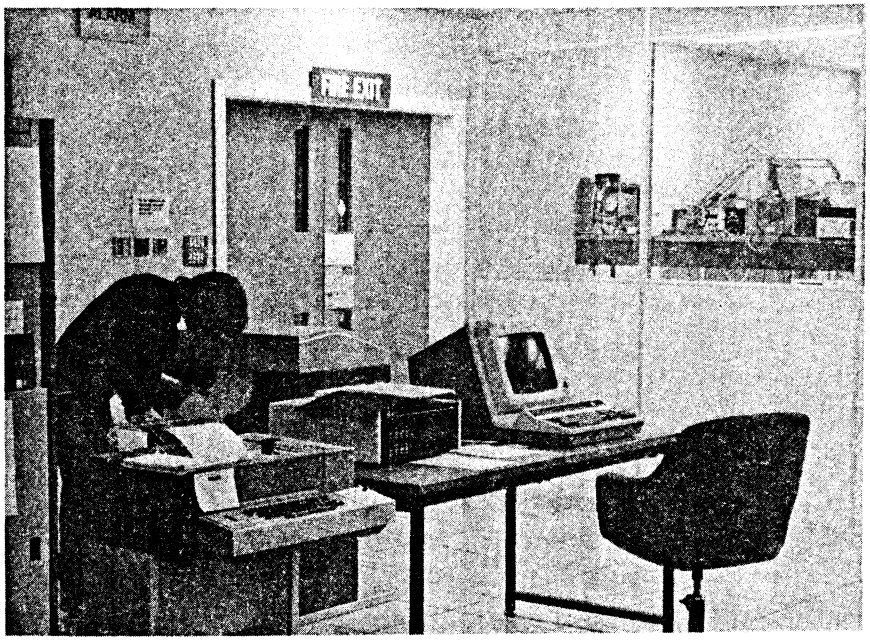
Safety.

T.Buckley.

Finance and Staff.

P.D.Schofield.

Diablo printer listing documentation for a new computer system.



Departmental file store, line printer and ICL 7502 editing station.

[CSR-8-77] L.Casey & N.Shelness. A domain structure for distributed computer systems. 1977.

[CSR-9-77] R.Milner. A theory of type polymorphisim in programming. 1977.

[CSR-10-77] C.Adams. An experimentally validated model of the paging drum. 1977.

[CSR-11-77] M.Gordon, R.Milner & C.Wadsworth. The Edinburgh LCF. 1977.

[CSR-12-77] J.C.Adams, W.S.Currie & B.A.C.Gilmore. The structure and use of the Edinburgh Remote Terminal Emulator. 1977.

[CSR-13-77] L.G.Valiant. Graph theoretic arguments in low level complexity. 1977.

[CSR-14-77] L.G.Valiant. The complexity of computing the permanent. 1977.

[CSR-15-77] L.G.Valiant. The complexity of enumeration and reliability problems. 1977.

[CSR-16-77] M.Gordan, R.Milner, L.Morris, M.Newey & C.Wadsworth. A metalanguage for interactive proof in LCF. 1977.

[CSR-17-77] D.Angluin & L.G.Valiant. Fast probabilistic algoritms for Hamiltonian circuits and matchings. 1977.

[CSR-18-77] N.Shelness, D.J.Rees, P.D.Stephens, & J.K.Yarwood. An experiment in doing it again -- but very well this time. 1977.

Additional Reports and Documents.

[1] R.Milner, L.Morris & M.Newey. A logic for computable functions with reflexive and polymorphic types. 1975.

[2] P.D.Schofield. Notes on IMP programming. 1975, revised 1977.

[3] H.Dewar, PDP-15 Users Guide. 1974.

[4] H.Dewar, ISYS Users Guide. 1975.

[5] H.Dewar, V.Eachus, K.Humphry & P.Mclellan, The File Store. 1977.

[6] H.Dewar, K.Humphry, P.Mclellan, F.Stacy & R.Thonnes, ISYS Systems & Utilities Guide, 1977.

Appleton Tower Users Committee.

F.Stacey.

Teaching, Learning and Assessment Committee.

P.D.Schofield.

Edinburgh Computer Users Committee.

J.Tansley, D.J.Rees, R.Thonnes.

Edinburgh Computer Committee.

S.Michaelson, J.Tansley.

ERCC Charging and Allocation Committee.

S.Michaelson.

Computer Equipment Panel.

S.Michaelson, J.Tansley.

Faculty of Science.

S.Michaelson, P.D.Schofield, R.Milner, D.J.Rees, A.S.Wight.

Senatus.

S.Michaelson, P.D.Schofield.

4.0 Undergraduate Degrees in Computer Science.

4.1 Introduction.

The following Honours Degrees are offered:-

(a) Honours in Computer Science
(b) Joint honours in:-

(i) Computer Science and Electronics
(ii) Computer Science and Mathematics
(iii) Computer Science and Physics.

About one third of the student's time is normally devoted to Computer Science in each of the first two years. The third and fourth years are wholly on Computer Science in case (a) above and divided between the two Departments concerned in the other cases.

The other courses taken in the first two years are determined by the intended Honours degree but their may be choices available. These can be discussed either before applying or more usually, with the student's Director of Studies on arriving at University. For example, Business Studies and Economics may be studied along with Computer Science. No previous experience is expected for Computer Science. Evidence of academic ability is more important than any specific background. Obviously for the joint degrees a student must have had suitable preparation in the subject to be studied along with Computer Science. Courses in other subjects are the responsibility of the Departments involved. In some cases the decision as to which Honours degree a student may wish to graduate with can be delayed by an appropiate combination of subjects in the first two years.

Undergraduate Admissions: A.S.Wight.

Course Enquiries: A.S.Wight.

General entrance requirements are set out in the University General Prospectus along with details of how to apply for admission. Further information about the Science Faculty and the B.Sc. degree is contained in the Science Faculty Prospectus.

General Enquiries:

Faculty of Science Office (Admissions)
University of Edinburgh,
West Mains Road,
Edinburgh, EH9 3JY.

N.A.Rose, Computer aided design of printed wiring boards. Ph.D., 1970.

P.C.Tavares, A graphic system to display physiological responses. M.Sc., 1974.

C.H.Whitfield, On the design of editors for small computers. M.Sc., 1972.

A.S.Wight, Backup, recovery and archiving of files in a multi-access computing system. Ph.D., 1974.

EMAS Reports.

[EMAS-R-1] H.Whitfield, A.S.Wight. EMAS, The Edinburgh multi-Access System. 1974.

[EMAS-R-2] D.J.Rees. The EMAS Director. 1974.

[EMAS-R-3] G.E.Millard, D.J.Rees, & H.Whitfield. The Standard EMAS Sub-System. 1974.

[EMAS-R-4] N.Shelness, P.D.Stephens, & H.Whitfield. The Edinburgh Multi-Access System, Scheduling and Allocation Procedures. 1974.

[EMAS-R-5] A.S.Wight. The EMAS Archiving Program. 1974.

[EMAS-R-6] P.D.Stephens. The IMP language and compiler. 1974.

[EMAS-R-7] J.C.Adams, & G.E.Millard. Performance measurement of the Edinburgh Multi-Access System. 1975.

Computer Science Reports.

[CSR-1-77] A.Anderson, M.A.C,Currie, L.Marshall, P.S.Robertson, & B.C.Wilkie. A User's guide to a PDP-16 design aid. 1974, revised 1977.

[CSR-2-77] G.Milne, & R.Milner. Concurrent processes and their syntax. 1977.

[CSR-3-77] D.Rees. DCAP; David's circuit assigment program. 1977.

[CSR-4-77] J.Reynolds. Programming with transition diagrams. 1977.

[CSR-5-77] R.Milner. Flow graphs and flow algebras. 1977.

[CSR-6-77] J.Reynolds. Reasoning about arrays. 1977.

[CSR-7-77] N.Saheb-Djahromi. Probabilistic non-determinism. 1977.

12.0 Departmental Publications.

Theses

C.V.W.Armstrong, Microprogrammed control of an associative processor. Ph.D., 1977.

T.Cadman, Design of printed circuit board layouts using graph-theoretic methods. M.Phil., 1974.

L.M.Casey, Computer structures for distributed systems. Ph.D., 1977.

B.I.Dervisoglu, Computer aided design techniques applied to logic design. Ph.D., 1973.

A.Freeman, A phrase structured macro scheme. Diploma, 1968.

B.A.C.Gilmore, IMP as a tool for small operating systems implementation. M.Phil., 1977.

D.Harverson, The Remes algorithm. Diploma, 1967.

A.K.Hope, The application of interactive computing techniques and graph-theoretic methods to printed wiring board design. Ph.D., 1973.

R.B.John, The design of systems for telecommunications between large and small computers. Ph.D., 1974.

J.M.Officer, The design and implementatation of a language for manipulating algebraic formualae. Ph.D., 1972.

C.J.Pavelin, The improvement of program behaviour in paged computer systems. Ph.D., 1970.

G.D.Ritchie Computer modelling of English grammar. Ph.D., 1977.

D.J.Rees, The design and implementation of programming languages for symbol manipulation. Ph.D., 1969.

4.2 Course Description.

Throughout the course emphasis is placed on practical experience with computers. From the first year onwards, Computer Science students have access to consoles connected to the University's large multi-access computer system, whose software was designed in the Department of Computer Science. From the second year onwards they also have "hands on" use of several of the Department's smaller machines and in the Honours years (third and fourth years) about one third of the work in the well-equipped laboratories centres around building systems from register-transfer construction kits.

For the third and fourth years in particular, hardware and software are brought together in the study of the design and construction of computer systems as a single entity. This is accompanied by a theoretical study of the subject which, though calling for mathematical reasoning, does not assume any specific mathematical knowledge other than that taught in the Computer Science classes. For those taking Honours in Computer Science (case (a) above), a choice of advanced topics is offered in the final year. Two thirds of the time is spent on the study of these and one third on an individual project.

For each course, there is a written examination in June. In determining the results, marks are awarded for this written examination and for course work submitted during the year. In 1st and 2nd years and 3rd year Ordinary coures, there is a re-sit examination in September. In 1st, 2nd and 3rd years, there are also class examinations in the December and/or March. These are principally to enable students to check their progress, but the results may be taken into account in the student's favour in the event of borderline results in June or September. Class examination results are also taken into account in awarding First Class or Second Class Merit Certificates in 1st, 2nd and 3rd years.

4.3 Computer Science 1 (1st year)

Course Co-ordinators: A.S.Wight. P.D.Schofield.

First Half Course.

This course consists of 36 lectures and associated practical work. After the first few weeks on a common course, students are advised to attend either an A or B course. The A course is the normal pre-requisite for entry to Computer Science 2. The B course is more suited to the non-specialist.

Both courses consist of:-

1. Introductory programming in IMP (similar to ALGOL 60) with an emphasis on the design of well-structured algorithms to manipulate numeric and non-numeric data.

2. Introductory data structures including lists, trees and some manipulation techniques such as sorting and merging.

3. Introductory computer systems.

Second Half Course.

This course which consists of 33 lectures and practical work, though not a pre-reqisite for Computer Science 2, is available to give a full Computer Science course unit in the first year. It is strongly recommended that students intending to enter more advanced courses attend. The course introduces concepts associated with operating systems, compiling techniques and file structures. More comprehensive programming exercises are also attempted.

All teaching of the first year courses takes place in the Appleton Tower located in the University's central site.

Computer Science 1 Examinations. (June and September).

There is a three hour written paper (half on each half course). Half the total marks awarded are for this paper and half for course work. There is also a practical examination, but those whose course work is satisfactory are exempted from this. A student who is awarded a First or Second Class Merit Certificate is given exemption from both written and practical first year examinations; these certificate are awarded on the basis of class examination results and course work.

11.0 Graduate Students.

K.Adam,	B.Sc., Computer Science & Elect. Eng. Edinburgh 1976.
A.M.Anderson,	B.Sc., Computer Science, Edinburgh, 1974.
I.Buchanan,	B.Sc., Computer Science, Edinburgh, 1975. (currently at Caltech).
C.Bradley,	B.Sc., Computer Science, Edinburgh,1977.
J.Carden,	B.Sc., Mathematics, Edinburgh, 1969.
A.Cohn,	B.Sc., Mathematics, Stanford, 1974.
J.Gorman,	B.Sc., Computing Science, Glasgow, 1977.
J.G.Grimson,	B.Sc., Elect. Eng. Dublin, 1970, M.Sc. Computer Science, Toronto, 1971.
W.A.Laing,	B.Sc., Computer Science, Edinburgh, 1972.
L.Legarreta,	B.Sc., Elect. Eng. Mexico, 1970, M.Sc. Computer Science, Wisconsin, 1971.
G.Lev,	B.Sc., Hebrew Univ. Jer., M.Sc. Feinberg Graduate School, Weiztman Inst. of Sci.
P.McLellan,	B.A., Mathematics and Computer Science, Cambridge, 1975.
G.E.Millard,	B.Sc., Mathematics, London, 1963.
G.J.Milne,	B.Sc., Computational Science, St Andrews, 1974.
C.T.Scott	B.Sc., Computer Science and Mathematics, Edinburgh, 1974.
M.Seysen,	Vor Diplom, Frankfurt, 1977.
A.Smailagic,	Dip. Elect. Eng., Sarajevo 1973, M.Phil., 1976, Sarajevo.
L.D.Smith,	B.A., Mathematics, Cambridge, 1974.
R.Thonnes,	B.Sc., Computer Science, Edinburgh, 1977.
C.H.Whitfield,	B.Sc., Psychology, Edinburgh, 1968, M.Sc., Computer Science, Edinburgh, 1972.
G.Winskel,	B.A. Mathematics, Cambridge, M.Sc., Mathematical Logic, Oxford.
W.G.Wood,	B.Sc., Computer Science, Edinburgh 1976.

9.0 Engineering Staff.

J.C.Dow,	Maintenance Engineer.
J.Johnstone,	Maintenance Manager,
P.J.Lindsay,	Design Engineer.
R.A.McKenzie	Design Engineer.

10.0 Office Staff.

Joyce Fisher; Heather Carlin; Dorothy McKie.

4.4 Computer Science 2 (2nd year).

Course Co-ordinator: R.Candlin.

This course, which involves about one third of a student's time, consists of 69 lectures and associated practical work. After common lectures for the first half session, students either take an A or B course. The A course is a pre-requisite for entry to Computer Science 3. The B course puts more emphasis on the application of computers and is the normal course for those who do not wish to proceed to Computer Science 3.

First Half Course. (36 lectures)

1. Familiarisation with a small single user computer, description of the computer at the machine code level, assembly code programming.

2. Introductory theory, concepts based on finite mathematics to include finite automata and combinatorial circuits and their complexity.

3. Introduction to compilers and compiling techniques, top-down and bottom-up syntax analysis, control of storage on a stack, implementation of a compiler for a simple language.

Second Half Course (A).

4. Introduction to an additional small multi-register computer, more advanced aspects of architecture including the programming of data input and output. (11 lectures).

5. Further theory, to include recursion equations as a simple programming language, some induction proofs of properties of programs, the relation between iterative and recursive programs, the general induction principle for well-founded orderings and the relation of one data structure to another. (12 lectures).

6. Register transfer design with PDP-16 macro-modules, use of register arithmetic units, computer and peripheral interfaces, hardwired micro-orders and micro-program control. (10 lectures).

Second Half Course (B).

This course has three equal components and, of these, computer graphics is compulsory, otherwise students may choose from either:-

Computational aspects of simulation and
Data processing
or
Computational aspects of simulation and
Register transfer design.
or
Further theory and
Data processing
or
Further theory and
Register transfer design.

Students following a main course in mathematics may find the further theory attractive while those in say, electrical engineering, may prefer the register transfer design course.

The syllabus for the Register transfer design and Further theory is the same as in 5 and 6 above. The contents of the other courses are given below.

7. Computer graphics. Various display devices such as CRT storage and refresh displays, graph plotters, and electrostatic plotters. Hardware for line primitives, display processors, instruction sets and character generation. The software aspects of device independence, transformations, clipping and windowing are also examined. Graphic languages and command languages along with interaction associated with light pens, rubber banding, zooming and dragging are considered. (11 lectures).

8. Computational aspects of simulation. Why simulate? Representaion on digital computers, limitations. Continuous v discrete systems, time domain analysis, frequencey domain analysis, state space models. Data models and probabilistic models. Simulation languages. (12 lectures).

9. Data processing. An introduction to COBOL. File definitions, reading and printing sequential files. System implementation, with pratical work on COBOL programming. Features of IDMS, data description facilities. COBOL programs within an IDMS environment. (10 lectures).

Computer Science 2 Examinations. (June and September).

There are two 2-hour written examinations. One third of the total marks are awarded for each of these papers and the remaining third for course work.

A.Q.Morton, M.A., B.D.; British Academy Librarian.
The study of literary style with special reference to problems of authorship and chronology.

M.Nielsen, LIC. SCIENT. Visiting Research Fellow.
Formal languages and automata theory. Theoretical aspects of computer science.

D.J.Rees, B.Sc., Ph.D., A.R.C.S.; Lecturer.
Design and implementation of multi-access operating systems. Programming language and compiler design.

P.S.Robertson, B.Sc.; Demonstrator.
Portable compilers. Systems software.

N.Saheb-Djahromi, B.Sc., Ph.D.; Visiting Research Fellow.
Non-determinism in semantics, probabalistic theory of computation.

P.D.Schofield, B.Sc., A.R.C.S.; Senior Lecturer and Head of Dept.
Special software for introductory teaching.

N.Shelness, B.A.; Lecturer.
Research into loosely-coupled computer systems.

F.Stacey, B.Sc.; Lecturer.
Systems software. Computer system simulation.

J.Tansley, B.Sc.; Lecturer.
Switching theory. Descriptive methods in the design and construction of digital systems. Digital methods in signal and image processing.

L.G.Valiant, B.A., M.A., D.I.C., Ph.D.; Lecturer.
Theory of Computation. Complexity of Algorithms.

A.Vernon, B.Sc.; Demonstrator.
Computer structures with special reference to micro-programming.

C.P.Wadsworth, M.A., D.Phil.; Research Fellow.
Theory of programming languages and computation. Formal semantics and its use in proofs about programs and languages. Lambda calculus models.

A.S.Wight, M.A., Ph.D.; Lecturer.
Performance monitoring and evaluation of multi-access operating systems. File system integrity.

8.0 Academic Staff.

J.C.Adams, B.Sc.; Research Associate.
Performance monitoring and modelling of multi-access operating systems.

D.Angluin, B.A., Ph.D.; Research Fellow.
Complexity of computation, algorithmic inductive inference.

A.Beaude, Diploma I.N.S.A.;Computing, Visiting Research Fellow.
Data base systems and evaluation of access algorithms.

T.Buckley, B.Sc., Ph.D.; Lecturer.
Design and construction of specialised micro-processor systems.

L.Casey, B.Sc., M.Sc., Ph.D.; Research Associate.
Research into kernel/domain structured systems.

E.R.S.Candlin, M.A., Ph.D.; Lecturer.
Introductory teaching and specialised micro-processor systems.

K.J.V.Eachus, B.A., M.Sc.; Demonstrator.
Software for small computers.

M.J.C.Gordon, B.Sc.; Research Fellow.
Semantics of programming languages. Formal specification of implementations and proof of their correctness. Applications of lattice theory.

J.P.Gray, B.Sc., Ph.D.; Lecturer. (leave of absence at CalTech until Jan 80).
Application of interactive graphical techniques to the design of integrated circuits. Graph morphology problems associated with layout.

K.Humphry, M.A.; Programming Assistant.
Software for small machines.

S.Michaelson, B.Sc., A.R.C.S., F.I.M.A., F.R.S.E.; Professor.
The study of literary style with special reference to problems of authorship and chronology. Linked ordinary and partial differential equations describing the behaviour of journal bearings under load. Queue related models of computing systems.

A.J.R.G.Milner, B.A.; Senior Lecturer.
Semantics of programming languages. Applications of mathematical logic to formalise the statement and proof of assertions concerning programs. Abstract models of computation, in particular, concurrent computation.

4.5 Computer Science 3 (3rd year)

Course Co-ordinator: J.Tansley.

This course, which is full time, consists of the following topics which examine many apects of computer science in detail. Each of the topics is listed below. Fuller details are given further on. Most of these topics have associated practical exercises.

		Lectures
1.	Digital Design	12
2.	System Components	12
3.	Digital Communications	12
4.	Control of Data Flow	18*
5.	Programming (Fundamental Concepts)	18*
6.	Programming (Compilers)	18
7.	Programming (Operating Systems)	18
8.	Programming (Applications)	18*
9.	Modelling and Systems performance.	18
10.	Analysis of Algorithms	18
11.	Semantics	18
12.	Computability	18
13.	Systems Architecture (by Seminar)	

Note Honours students attend all the above; Ordinary degree candidates may omit certain topics depending upon the examination papers they intend to take. The courses marked "*" are still being formulated and may be subject to revision.

Digital design (12 lectures).

Combinatorial logic comprising of switching algebra, minimization, universal logic arrays, generalized switching functions. Sequential logic comprising of the synthesis and analysis of synchronous and asynchronous cicuits. Races, cycles and hazards. Discussion of design with standard MSI and LSI components. Fault diagnosis of logic circuits.

System components (12 lectures).

The influence of technology on system components viz. semi-conductor developments. The differences between n-mos, p-mos, c-mos, bipolar etc.. Levels of integration, MSI and LSI, chip density, comments on physical limits. Computer aided design methods and construction techniques. Cost of construction. Noise and electrical transmission problems. Semi-conductor aspects of memory and processors, core memory. Peripherals; I/O terminals e.g. plotters, line printers, along with comparative cost and performance; Mass storage devices e.g. disc and magnetic tape, formats, controllers, comparative cost and performance.

Introduction to Digital Communications (12 lectures).

Communications theory, digital communications, parallel and serial forms, codes, errors etc. Parallel synchronisation, handshakes etc,. Serial synchronization, bit synchronization, frame and character synchronization. Packet formats and protocols, polled, asynchronous, half duplex, full duplex, BSC, HASP ML, HDLC, DDCMP. Packet network control, congestion, flow control, error control. Network services, high level protocols.

Control of Data Flow. (18 lectures).

Computational schemata, program graphs. Asynchronous modular systems, control features of RT systems. Various RT languages, control flow and various token flow models. Petri-nets and GMC, unification by vector addition systems. Multi and parallel process implementation. Horizontal micro-programs, architecture for micro-programming. Encoded and decoded control, encoding micro-programs. Micro-programming languages. Instruction sequences, gate signals and specification. Vertical micro-programming. Comprison of cost and performance of RT and micro-programming implementations.

Programming (Fundamentals). (18 lectures).

The details of this topic have yet to be formulated.

Programming (Operating Systems). (18 lectures).

Review of batch processing system programs, their components, operating charateristics, user services and their limitaions. Implementaion techniques for parallel processing of input/output and interrupt handling. Overall structure of multi-programming systems consideration of multi-processor configurations. Details of addressing techniques, store management, file system design. Interprocess communication, design of system modules and interfaces.

Programming (Compiler Construction). (18 lectures).

Review of programming language structures, translation, loading, execution and storage allocation. Compilation of simple expressions and statements. Organisation of a compiler including compile-time and run-time symbol tables. Lexical scan, syntax scan, object code generation, error diagnosis. Object code optimization techniques and overall design. Use of compiler writing languages. Bootstrapping.

PDP 11/20

Hardware:- PDP 11/20 processor; 40K bytes of core store; Interface to ICL 2970; Teletype console.

Use:- Communications link to ICL 2970.

PDP 11/10

Hardware:- PDP 11/10 processor; 32K bytes of core store; 16 line terminal multiplexor; HDLC data communications to EMAS; Centronics line printer; 12 Olivetti consoles; 4 Newbury VDUs.

Use:- Concentrator for consoles in the Appleton Tower that are linked to EMAS.

Additional Facilities.

These include a Calcomp 563 plotter; Tektronics terminals type 4014, 4010,4012 and a graphics tablet. There are also two Interdata 7/16 processors available for general experimental use.

Batch

Hardware:- Interdata 70; 64K bytes of core store; Olivetti console; data link to File Store.

Use:- Dedicated batch processor; compilation, text processing etc.

ICL 7502

Hardware:- ICL 7502 processor; 12K 16bit words of core store; Visual Display Unit with 128 character set; data links to file Store.

Use:- Sophisticated editing and File Store interrogation.

ISYS (North)

Hardware:- Interdata 70 processor; 64K bytes of core store; Diablo 30 Dual disc drive with 5 megabyte capacity; Teletype console; data links to the File Store; four satellite Interdata 74 processors each with 8K bytes of core store; Olivetti console switchable between the satellite and the main machine; data links connecting the satellite and mother machine.

Use:- Third and fourth year teaching; general programming and research.

ISYS (South)

Hardware:- Interdata 70 processor; 64K bytes of core store; Diablo series 40 disc with 10 megabytes capacity; Teletype console; data links to the File Store; four satellite Interdata 74 processors, each with: 16K bytes of core store; Olivetti consoles switchable between the satellite and the main machine; data links connecting the satellites and the mother machine.

Use:- Third and fourth year teaching; general programming and research.

File Store

Hardware:- Interdata 70 processor; 64K bytes of core store; 9 pairs of data links; Data Products 2260 line printer; Olivetti console; Two CDC 9762 80 megabyte storage modules attached via a Systems Industries controller.

Use:- General file support to machines in the Department.

Programming (Applications). (18 lectures).

Computer graphics are used to illustrate a number of basic problems in the design and implementaion of application system. The following topics are covered; systems overview as a user, languages for graphic description, command languages. Data organisation, file structures and data bases. Graphic processors and graphic I/O devices. Driving graphic devices, pseudo-display files etc,. Specific graphic functions, i.e. hidden line removal, co-ordinate transformations. Performance and use of specialised hardware.

Modelling and Systems Performance. (18 lectures).

Part A Application of probability to computer systems, simple queueing theory, Markov chains, state diagrams, forward-backward equations, steady states, birth and death processes, Little's theorem, Kintchine-Pollaczek equation, network models, decomposability, Buzen's algorithm, diffusion approximation.
Part B Simulation modelling, performance measurement of real systems, workload charaterisation, synthetic workloads, bench marks, performance evaluation and prediction.

Analysis of Algorithms. (18 lectures).

Introductory concepts. Algorithms based on splitting. Recurrence relations. Comparison problems, analysis of sorting, merging and selection problems. Information theoretic lower bounds. The class NP. Reducibilities among problems. NP-completeness. Fast algorithms for multiplying integers and polynomials. The finite Fourier transform. Matrix problems, algorithms and reductions, (e.g. Boolean transitive closure, shortest paths, determinates).

Computability and Formal Languages. (18 lectures).

Introduction, Cantor's diagonal argument, Turing's thesis. Turing machines, partial recursive function, TM techniques, variations on TMs, simulation arguments. Other models of computation, stack machines, counter machines, queue machines, equivalence to TMs. Universal Turing machines. Undecidabiltiy, halting and other problems, recursive and recursively enumerable sets. McCarthy recursion equations, operational semantics. Chomsky hierarchy, grammars versus machines, closure operations, normal forms, decision processes, parsing. Type 0 languages, context sensitive languages, regular languages, CF grammars, parse trees, Chomsky normal form. Push down automata, equivalence problem. Parsing, Younger's method, LL(k) parsing.

Semantics. (18 lectures).

Operational semantics of recusion equations and of simple imperative programming language P, the approach and proof of their equivalence. Data domains and function domains under complete partial order, monotonic and continuous functions, the least fixed-point operator, and its application in solving recursion equations. Proof of properties of recursively defined functions using least fixed-points. Denotational semantics of the language P, using complete partial order domains, equivalence with operational semantics, proof of properties of language P, use of continuations, extensions to P (including declarations and procedures) and their semantics.

Systems Architecture. (9 seminars).

Input and output organisation, handling of simple devices, intelligent devices e.g. FEP's, file store, etc. Storage organsiation, one level storage, storage hierarchy, caches, cost-performance, capabilities, addressing schemes. Processor organisation, pipelines, stacks. Inter-system communication, buses, multi-processors. The above concepts are illustrated with examples taken from IBM 360 & 370, DEC PDP-11 and VAX 11/780, CDC 6600/7600, Burroughs 5500/6500, ICL 2900. Reference is also made to other systems.

Computer Science 3 Examinations (Honours June, Ordinary June and September).

(a) Honours. There are three 3-hour written examinations, which contribute two thirds of the total marks, the remaning one third comming form coursework. For Honours, a student must pass this examination in June, at the first attempt.

(b) Ordinary. An Ordinary pass is awarded for satisfactory performance on any two of the three papers and corresponding course work.

Table 1

Departmental Computer Systems.

PDP 9 System

Hardware:-	PDP 9 processor; 8K of 18 bit word core store; 3 TU55 DEC tape transports; paper tape reader and punch; Olivetti console; Logabax printer; Tektronix 611 scope; data link to File Store
Use:-	Second year teaching; general programming.

PDP 15 System A

Hardware:-	PDP 15 processor; 24K of 18 bit word core store; 4 TU56 DEC tape transports; paper tape reader; Olivetti console; Logabax printer; data link to File Store
Use:-	Second year teaching; general programming.

PDP 15 System B

Hardware:-	PDP 15 processor; 32K of 18 bit word core store; 1 TU56 DEC tape transport; 1 TU10 magnetic tape transport; VT15 graphics processor; 16 Channels of ADC and DAC; floating point processor; fixed head disc with 512K word capacity; paper tape reader; data link to File Store.
Use:-	Second, third and fourth year teaching. Graphics projects and research work.

PDP 11/40

Hardware:-	PDP 11/40 processor; 64K bytes core store; one RK05 disc; Ampex 960 disc drive and System Industries Controller; DEC Writer II console; data links to the File Store and EMAS
Use:-	SRC funded research into multi-access systems performance.

Legos.

Hardware:-	Interdata 70 processor; 64K bytes of core store; Olivetti console; Tektronics 4002 terminal; data link to File Store.
Use:-	Research, development of hardware construction aids and hardware design languagues.

file storage immediately accessible on-line.

During the next year (1978) it is also planned to bring into service an ICL 2970 also running a locally written multi-access system. The kernel structure and data communications were devised and implemented by members of the Department. At present the 2970 has 1M byte of store, two 6M byte fixed head discs and four 100 M byte exchangeable discs.

There is also a comprehensive communications network both within the University and with remote sites. This allows access to a number of special peripherals such as a Versatek printer, a Calcomp plotter and a Free-scan digitzer.

The second branch of service is the batch mode of operation. The ERCC is bringing into service an ICL 2980 which has been provided for Edinburgh, Glasgow and Strathclyde Universities jointly. This latter computer is believed to be the most powerful in Scotland at the time of writing. It has 2 and 1/2M bytes main storage, 18M bytes drum storage, and 1600M bytes disc storage. The 2980 is the top model in ICL's 2900 range.

Programming Languages

The most widely used language in the department is IMP, an ALGOL-type language developed as the software implementation language for EMAS which has since come into general use throughout the University. The other main languages are FORTRAN, ALGOL-60, LISP, COBOL and PL/1 which are all available on one or more machines.

4.6 Computer Science 4. (4th year)

Course Co-ordinator: R.Milner.

Project Co-ordinators: L.Valiant, S.Michaelson

One third of a student's time is devoted to an indvidual project that is selected from a list provided by Departmental staff. The remaining part of the course is made up of a number of options from which a student chooses about seven. These include:-

1. Artificial Intelligence.
2. Commercial and Industrial Data Processing.
3. Comparative Programming Languages.
4. Computational Complexity.
5. Computer Aided Design.
6. Computer Systems Performance Evaluation.
7. Advanced Graphics.
8. Hardware Structure of Processors.
9. Computer Networks.
10. Operating Systems.
11. Semantics of Programming Languagues.

Artificial Intelligence. (12 lectures).

Introduction. Programs which solve problems in domains such as game playing, theorem proving or robot planning will be used to illustrate the different methods adopted. For example programs by Samuels (draughts) and Greenblatt and Berliner (chess) will be examined. Similarly the methods used in Slagle's integration program, Gelenter's geometory program and the program-correctness program of Boyer and Moore will be evaluated. The course will also look at methods used in pattern recognition where the approach taken is to combine prior knowledge of a class of scenes with symbolic knowledge of a given scene to produce a structural description of the input picture.

Commercial and Industrial Data Processing. (12 lectures).

Introduction to the problems which arise in commerce and industry. Influence these have on computer systems design. A group project evaluating the computational needs of one of a list of organisations, e.g. Health Centre, Brewery, Library etc,.

Comparative Languages. (12 lectures).

This course attempts to provide a framework for the comparison of languages by considering each language at three levels, that is, in terms of its structure, its utility in solving problems and its target implementation. The languages COBOL, FORTRAN, ALGOL 60, SIMULA, PASCAL and LISP are used as examples.

Computational Complexity. (12 lectures).

Multi-tape Turing machines, time and space as resources, relationship with other models, one-tape versus k-tapes, lower bounds for one tape, non-deterministic machines. Time and space complexity classes, hierarchies, linear speed ups, easy simulation results, (P,NP,PSPACE,NPSPACE). Complete problems for NP, PSPACE, EXPTIME, EXPSPACE. Relations between classes, time versus space, DSPACE versus NSPACE. Lower bounds for finite problems.

Computer Aided Design. (12 lectures).

Graph based system representation e.g. Chartware, Logos, and language based representation, e.g. RT languages. Simulation, RT-level, gate level, event driven, table driven, three valued. Electronic layout algorithms for partitioning, placement and routing. Non-algorithmic design systems and the use of graphics. Data structures and programming techniques for C.A.D.

Computer System Performance Evaluation. (12 lectures).

Performance measures, load measures, measurement techniques, work load characterisation including program behaviour and benchmarks, simulation and experimental techniques, modelling including regression analysis, mathematical and simulation models, validation, case studies, overhead analysis.

Advanced Computer Graphics. (12 seminars).

The course consists of a series of seminars that co-ordinate a group design and implementaion of a system to cope with a specific graphical problem, e.g. animation, image analysis or picture drawing.

Hardware Processors. (12 lectures).

The course aims to describe some of the benefits provided by the use of current LSI systems components in processor design and to develop some techniques for practical design problems. Asynchronous State Machine (ASM) chart techniques will be introduced to link micro-program control structures with more familiar register transfer flow structures. Micro-programmed and programmable logic arrays will be compared. Special purpose processors for digital filters and particle physics data analysis will also be described.

Networks. (12 lectures).

This course consists of a critical review of the current state of computer networking. The focus is on the European Informatics Network, although students will be expected to familiarize themselves with other networks though a course of independent reading.

7.0 Computing Facilities

7.1 Departmental Facilities

The Department's computing machines consist of a number of mini-computers that have been purchased over the past years. Most of these machines are connected together by high speed serial data communications links. The data link, which was designed and constructed in the Department, operates at either one quarter or two megabaud. Several of the computers are connected together to form clusters that are used for teaching and research, the software for which was also designed and implemented in the Department. The full complement of machines along with their principal teaching and research commitments, is shown in table (1)

Central to many of the systems in the Department is the File Store. The software, again designed and developed in the Department, is implemented on an Interdata 70 that has 135 megabytes of exchangable disc storage. The file store also spools output for a Data Products lineprinter and a Diablo Hytype document printer.

Most of the Departmental systems offer a number of programming languages (IMP,HAL,ALGOL) and various utilities. For example there are editors and document preparation programs. Documentation on most of the languages, operating systems and utilities is available and listed in section 12.0.

Much of the Department's hardware teaching is based on register-transfer modules (RTM). These modules which can be assembled into a backplane system have evolved from those devised by the Computer Science Department at Carnegie Mellon University. The latter are now sold by the Digital Equipment Corporation as PDP-16 kits. We now have a wide selection of these modules including some which can be micro-programmed. Quite comprehensive systems can be built with these modules which are ideal for quick experimental investigations. Various software aids are available to assist design and construction with these kits.

Micro-processor systems are also being evaluated and in particular the Texas 990 and Signetics 2650.

The Department has a well equipped electronics workshop which contains the usual selection of hand tools and electronic measuring devices. These include Tektronix scopes and a Hewlett Packard logic state analyser. A number of software aids are also available to generate layout for printed circuit boards and wire wrapping on a semi-automatic machine.

7.2 ERCC Facilities

The Edinburgh Regional Computing Centre (ERCC) provides the main computing service for the University and the Department makes full use of this.

There are two main branches of the ERCC service. First, there is a multi-access service using EMAS on two ICL 4-75 computers. Each 4-75 has a substantial configuration including 1M byte of core store, 6M bytes drum storage and 700M bytes disc storage . The disc store in particular allows each accredited user to maintain a useful amount of

ii) Concurrent processes.

A formal description of concurrent processes is being developed in terms of flow graphs and flow algebras. Using a primitive notion of synchronised communication, mathematical models have been developed that allow a precise examination of concurrency. The relationship between these concepts and programming languages is also being studied.

iii) Complexity of Computation.

This work is investigating:-

a) lower bounds on the number of operations required for the solution of specific problems,

b) algorithms for the solution of problems with 'good' upper bounds on the number of operations required,

c) the effect on the efficiency of computation brought about by variations in sequencing, the introduction of parallelism or the relaxation of conditions for exact solutions,

d) and the relative complexity of classes of problems with respect to lower bounds on computation time and space.

Aspects of this work are supported by a SRC grant.

6.3 Computer Structures.

Research in this area is investigating the fundamental nature of computer structures as they relate to both the computational process and its implementation in hardware. Two lines of activity are being pursued.

i) Descriptive methods and design languages.

In examining descriptive languages we are using methods emanating from computational complexity to devise notations for design that allow only feasible procedures. That is, we are considering what restrictions need be placed on a designer so that the properties of his design may be automatically evaluated for testability, correctness, etc. To this end a design-simulation system is being developed with due consideration being given to eventual implementaion in hardware. Complementary to this, basic hardware primitives, either as micro-code or individual components are also being studied.

ii) Design with Very Large Scale Integrated Circuits

This programme of research is directed towards producing computer aids to assist in integrated-circuit design. The project has produced design software in collaboration with the Wolfson Micro-electronics Liasion Unit. A version of this is now in use in industry. Present developments include the development of mask-checking to detect design rule violations and the evaluation of functional performance of a design. Longer term aims include the automatic layout of integrated circuits.

Operating Systems (12 tutorials).

This option consists of one major programming execise supported by tutorial discussions. Each student individually implements a multi-programming operating system for an Interdata 74 designed to run a batch of programs supplied by a second Interdata. Each program will have varying characteristics such as size, CPU-time needed and I/O activity. The intention of the option is to allow students to put into practice some of the concepts and methods presented in the formal lecture courses.

Semantics. (12 lectures).

Reasons for the precise description of computer languages and their implementation. Early methods of semantics i.e. Burstall, Floyd, Knuth, Landin and Strachey. Denotational semantics, abstract syntax, semantic function, semantic domains, environment, store, various kinds of continuation (e.g. command, expression declarations), various kinds of values (e.g. denotable, storable expressions etc). Styles of semantics and alternative ways of handling features such as subroutines and input/output. Operational semantics, methods of formal description of implementations, VDL, machine orientated semantic equations (e.g. stack and store semantics), reduction rules, evaluation mechanisms (e.g. SECD machines). Axiomatic semantics, the explication of language features by inference rules, advantages and disadvantages. Introduction to Hoare's work, relation to other approaches, Ungler's result.

Computer Science 4 Examinations. (June only).

There are two 3-hour written examinations, which together contribute one third of the total marks, one third comming from the student's indivdual project work and one third from course work associated with the options.

Final Honours Assessement.

First, Second or Third Class Honours may be awarded. For Honours in Computer Science, the results of the Computer Science 3 (Honours) and Computer Science 4 examinations are given equal weight. For joint Honours degrees, the results of Computer Science 3 (Honours) and the corresponding examination in the second Department concerned are given equal weight.

5.0 Postgraduate Degrees in Computer Science

It is currently only possible to study for research degrees viz. M.Phil. and Ph.D. in the Department. The Diploma course which used to be offered is no longer available nor do we offer a one-year course leading to the M.Sc. degree. The normal periods of registration are two years for the M.Phil. and three years for the Ph.D.. Part-time study is only available if the candidate is either a member of staff of the University or of an 'Associated Institution', or an Edinburgh graduate. Candidates are normally registered in the category of 'Supervised Post-graduate Student' for the first year of study, prior to transfer to the degree course considered appropriate, with back-dating of registration.

Financial Support

U.K. students may be eligible for Science Research Council Studentships. The value of a studentship depends on age, experience and family circumstances. A first-class or upper second-class degree is virtually an essential qualification. The Department usually receives a quota of studentships each Spring to which students can be nominated.

The University has a small number of postgraduate awards open to a wide class of applicants including those from overseas. Competition for these is extremely keen. Details can be obtained by writing to the Secretary to the University.

Applicants from Commonwealth countries may be eligible for Commonwealth scholarships. Details of these can be obtained from the High Commission in the student's own country.

The Department does not have any Demonstratorships or Teaching Assistantships to offer students.

Visits to the Department

Candidates already in the U.K. are encouraged to visit the Department, both to see research in progress and to discuss particular opportunities. Please contact Dr.D.J.Rees to arrange a visit.

Application Procedure

Students wishing to be considered for entry as postgraduate students, should complete an application form (available from the Department), and return it as soon as possible. When the form is received it will be considered with a view to suitability of the candidate and choice of potential supervisor on the staff of the department. If the candidature is regarded as suitable, a University application form will be sent to the student for completion and return. This form is then transmitted to the Faculty of Science Postgraduate Studies Committee for consideration. The process of consideration may take up to two or three months depending on timing of meetings and possible further enquiries such as contacting referees.

6.0 Research Activities of the Department

The Department covers a wide range of research interests with a special emphasis on the design of computer systems, the theory of computation and computer structures. The individual research interests of the members of staff are described in section 8.0.

6.1 Design of computer systems

Current research activity in computer systems is concerned with the development of distributed computer systems and the performance evaluation of multi-access systems.

i) Distributed Computer Systems.

Previous work on general purpose multiple instruction and multiple data-stream computer systems has mainly concentrated on multiple processor systems in which a number of processing units execute code and manipulate data held in a common shared memory. Research work now being conducted in the Department is concentrating on multiple computer systems in which processors access only their own private memory but still function as an integrated system. Investigations are proceeding as to how this form of decentralized control can operate so that code, data and computations can be dynamically moved from processor to processor to gain improved performance. This work is supported by SRC grants.

ii) Performance Evaluation of Multi-Access Systems.

Starting with the Edinburgh Multi-Acess system (EMAS), where system behaviour can be measured, the measurement of interactive user behaviour leads to the construction of synthetic work loads and subsequently to the measurement of system behaviour in a controlled environment. This includes the study of locality in the levels of storage hierarchy. Work is also proceeding on the construction and validation of models of EMAS using the above measurements. One possible use of these models will be an investigation of adaptive scheduling policies. This work is supported by a SRC grant.

Much of the above work stems from the Department's involvment in the design of EMAS which now forms the main interactive computing facility for the University. There is a strong interest in this work in relation to improved implementation on other computers.

6.2 Theory of Computation.

Three areas are currently being studied;

i) Logic of Computable Functions.

This work is concerned with the precise mathematical formulation of problems in computation such as correctness of programs (including compiling), and the methodology of conducting proofs of appropriate theorems by interacting with the computer. For this purpose, a metalanguage has been designed in which partial or complete proof strategies may be programmed. The system has been fully implemented on a DEC System Ten. This work is being supported by a SRC grant.