COMPUTER ENGINEERING AND DESIGN

F. H. Sumner and D. J. Kinniment

Rapporteurs: Mr. T. Anderson
Dr. A. J. Mascall

Abstract: An attempt is made to explain the attitudes of Manchester University to the teaching of computer design.

The course at Manchester contains a considerable amount of computer engineering, and covers in detail the electronics etc. of logic elements and the different types of store. With this background, teaching can be taken from the engineering design level to the features visible to the user.

Through all the design teaching, great stress is placed on relating the effect of the various developments in computer engineering and design on the systems programmer and other users of the complete computer system.

1. Historical Survey of Computing at Manchester University

In order to give a general impression of the Manchester Computing Science Department, a brief history of its development will be useful.

The work on computing started in 1948 under the direction of Professors Williams and Kilburn. This resulted in a prototype machine in 1949, which was developed commercially by Ferranti (1951). Work then continued with MEG, the first British computer to have floating point facilities, which led to the Ferranti MERCURY in the mid-fifties. The cooperation with industry continued and resulted in the building of ATLAS in 1962. This theme of designing and building computers and their software, followed by their commercial exploitation, has influenced both the research and teaching activities of the department.

By the mid sixties computing science at Manchester has grown away sufficiently from its parent, Electrical Engineering, and became an independent department in 1964. The first undergraduate intake was in 1965.

2. Current Situation and Future Trends

		Present	Projected
Staff	Academic	27	35
	Other (technicians etc.)	33	71
Students	Undergraduate	170	240
	Post-graduate	33	50-60
	External to Department	500	1000-1500

The above table gives approximate figures for the different categories of staff and students at the present time (1971), and as anticipated for the period 1972-77. Considerable benefit is obtained for the department from the extensive cooperation with industry, which makes possible a type of research that could not otherwise have been undertaken in a purely academic environment.

In the area of postgraduate study the intention is to increase the number of students taking an M.Sc. degree; a decrease is expected in the proportion continuing to Ph.D.

Short courses on computer appreciation, design and programming are given for students throughout the University.

The courses given in the department have developed considerably over the years, and have confirmed the opinion that to teach Computing Science (which at Manchester means computer design) it is essential to integrate the hardware and software aspects of the subject, emphasising their close relationship throughout the degree course.

3. Outline of the Undergraduate Syllabus

The entrance requirements for undergraduates are A-level Mathematics and Physics. Because of the demand for places it has been possible to maintain a uniformly high standard of student intake and it is felt that this situation will continue even after the projected expansion.

Referring to figure 1, the discussion here will be concerned mainly with the two left-most columns: the remainder of the timetable is devoted to the programming and mathematical sections of the course. As can be seen, there is a large amount of laboratory work in each of the three years, reflecting the view that the subject is practical in nature. For the first two years no options are permitted; the intention is that even if a student chooses, in his third year, to specialize in software (as $\frac{2}{3}$ do) he will have obtained sufficient knowledge of hardware to enable him to communicate with computer engineers. In this respect it is hoped that the students will be able to listen to, and understand, specialist arguments from either side of the hardware-software divide. In their third year all students take the courses on Computer System Engineering and Computing Techniques and Operating Systems, which are packed into the first six weeks. These are designed to help the student select three optional subjects which are studied for the rest of the year. In addition, all students must complete a large project requiring roughly 250 hours of laboratory work.

At this point it is appropriate to mention the various outlets that graduates have found on leaving the University. Most work on software, the majority going to manufacturers, but some to software houses or user environments. Many have said that their University training has been of considerable advantage to them, and this is borne out by their progress in these establishments.

4. Computer System Design Courses

The first year course on this subject, Computer System logical Design, is given in two sections, the first by Professor Kilburn, the second by Professor Sumner. The topics covered by Professor Kilburn range from simple circuit analysis to diode and transistor logic. Students are quickly introduced to logic and the implementation of logical elements so that they realize that computer design is not merely a matter of writing down a logic diagram. Construction of a simple adder and adder-subtractor leads on to a detailed study of the design of a small machine, the machine in question being the original Manchester computer. This can be completely described in the space of a few lectures, which could not be done for more modern small machines, e.g. the PDP-8 or Modular-1. It is also possible to stress the timing controls involved in this computer's operation, with the intention of instilling an appreciation of the importance of timing considerations in general. Throughout all courses an attempt is made to indicate historical developments and to show how innovations arise from the efforts required to solve particular problems.

The second part of the course begins with the study of different number representations and continues with a review of the techniques of parallel arithmetic and floating-point arithmetic. These topics are illustrated by the consideration of arithmetic units built by the department. The policy of describing examples of work done in the department is regarded as beneficial, since some of the enthusiasm of those involved is conveyed to the students. Six lectures are spent in discussing formal logical design, again from a practical viewpoint, so as to provide sufficient background for a small design experiment in the second year.

On reaching the third year the students are already familiar with the basic building blocks, such as different types of memory and peripherals, and are aware of software techniques such as compiler organisation. The final year design course covers the problems which arise in the design of large computer systems. A recurring theme is the problem of the transfer of information within a system: this is analagous to the problems of energy transfer which occur in the physical sciences. The following systems are referred to as illustrations of how various design problems have been solved:-

1. ATLAS system
2. MU5 research machine at Manchester
3. MULTICS system on the GE645 (especially virtual memory)
4. IBM 360/67 (mainly for the paging aspects)
5. IBM 360/85, 360/195 (techniques for maximising throughput)
6. CDC 6600, 7600 and STAR (problems of data transfer)

Once again, the Manchester machines are described because of the intimate knowledge held by the department, as opposed to the second-hand information available about the other commercial systems.

5. Electronic Engineering Courses

The basic components of the undergraduate computer science course given at Manchester are the following:-

1. Study of computer applications.
2. The basic principles of system software which enable the computer to be used efficiently.
3. The engineering of the hardware which enables the best use of currently available technology in a system.
4. Study of the combination of hardware and software.

The courses described here are concerned mainly with aspects 3 and 4 above. Care is taken to impress on the student that other technologies than electronic may in future be available, but naturally current teaching must include a thorough treatment of electronic engineering in order to illustrate the practical design of present-day systems. The intention is to foster an attitude such that the student is able to cope with problems presented by experts, to understand the constraints imposed by technologies, and to evaluate specialist arguments.

In the first year the main aim is to ensure that the basic elements of electricity, magnetism and electronics are known by the undergraduates. The following topics are covered:-

Resistance
Networks
Fields
A.C. Theory
Semiconductors
Transistors
Cathode ray tube
Simple machines

Emphasis is placed on applications to, for example, core storage devices and D.C. motors as used in tape drives. The corresponding first year laboratory course consists of a series of formal experiments occupying two afternoons per week. These experiments are designed to demonstrate:-

(a) Device characteristics - magnetic cores and transistors.

(b) Simple circuit characteristics - response to transients in R-C, R-L circuits.

(c) Common circuit configurations - diode gates, amplifier circuits including chaining limitations.

(d) Simple logical design - binary counter, serial arithmetic using TTL elements.

(e) Measuring equipment - use of oscilloscopes and meters.

In the second year the course on memory systems and logic circuits breaks down as follows:-

MEMORIES	Memory system costs and performance. Surface recording. Core memories. Metal oxide silicon memory.
LOGIC	T.T.L. circuit family. Arithmetic Control Interconnections

Details of the engineering compromises needed for the practical design of core memories, drums, disks and tapes are given. The discussion of logic circuits includes comment on practical problems such as cabling costs and efficient layout, and not merely the minimization of the logic elements.

The electronic circuit engineering course, given in parallel with the above, also consists of two parts:-

LINEAR	The transistor Design of a linear amplifier Negative feedback Semiconductor Physics
DIGITAL	Charge control concept Waveform shaping Two state circuits Timing circuits

The laboratory course for the second year involves the selection of either a set of formal experiments or a design project. The formal experiments are grouped into the following three categories:-

(a) Complex circuit concepts - negative feedback and timing circuits.

(b) Dynamic component characteristics - heat effects on semiconductors, frequency response.

(c) Demonstration systems - core and drum store, A-D converter.

The design projects available illustrate either logical or circuit design, and include a small computer control, variable radix counter, variable width pulse generators and a differential amplifier. This course provides an opportunity for the study in detail of selected aspects of component behaviour, for instance heat effects, variation of drive currents and various tolerances.

The compulsory computer system engineering course in the final year, which runs concurrently with the system programming course, discusses in general terms the problems which arise in the planning and development of large computing systems. Individual topics discussed are:- the specification of the machine's performance from initial discussions, cost-performance tradeoffs, reliability considerations, choice of control organisation (synchronous, asynchronous, microprogrammed), communication between groups and means of documentation, use of simulation techniques and the maximization of hardware efficiency.

There are two optional third year courses on computer engineering. The hybrid computers and control systems course deals with the hardware and software of hybrid devices such as analog-digital converters, and their applications, for example, the design of a simple flight simulator. The characteristics and design of linear feedback systems are described, and an introduction given to data transmission - communication channels, modulation, filters, features of practical systems. The course on advanced electronics covers the following subjects:-

Integrated circuit technology
High speed logic circuits
Metal oxide silicon devices
Transmission lines
Magnetic logic
Advanced memory techniques - Plated wire and integrates circuit memories
Graphical and visual displays

In keeping with the emphasis placed on the practical nature of this degree course, the final year laboratory work consists of four hours per week spent on a substantial project. The design of a small computer system (based on a cut-down MU5 order code), construction of a highway control system to be linked with a PDP-8, and the investigation of a plated wire store are typical recent third year projects. Currently a printed circuit facility is provided by the department, enabling student designed systems to be made up by technicians. Future third year projects may assist in automation of this facility with a software controlled draughting device.

6. Overview

Throughout the student's stay in the department, the approach adopted towards his education is essentially that of a master-apprentice training. The methods by which present-day computing systems have been designed and built are explained, and the extent of their success, and their short-commings are discussed. The "Black Box" approach is carefully avoided, and with reference to Professor Wheeler's lecture, details of the type described there are given due attention. In the same vein, the provision of parity checking to give one bit error detection for core storage is described, to show that increased reliability can only be bought at the expense of extra hardware. Students should develop an engineering attitude to the problem of achieving a balance between cost, reliability and speed. They are taught to appreciate the differences and similarities between hardware and software, and to decide when one technique should be used in preference to the other. An example of this is the use of software for page replacement algorithms while such algorithms are being evaluated, and then, when a correct choice of method can be made, its implementation as hardware. The point is made that information is required before decisions can be taken, and that obtaining such information can be costly, both financially and in time. Thus there is a need for making the best decision in the time available, as is demonstrated by consideration of the transfer of information between hierarchies of storage. When a page must be returned from core memory to a drum, there is time to indulge in a complex programmed algorithm to select the page to be removed, but if information is to be removed from an associative memory, the algorithm to do this must be hardware implemented, fast and simple. At Manchester, a "hit rate" of 96% on a 24 word associative memory was upgraded to 98% by provision of a hardware removal algorithm; at the same cost a 32 word associative memory could be built with a 99.5% hit rate, showing that the cruder technique can sometimes provide the best solution.

The effect that user and system programmers can have on a computer's performance is considered; control transfers give an example here. The production of very compact code often results in a high proportion of control transfers, and in any case, branching instructions tend to occur at a rate of about one in five. At the hardware level, the problems raised by prefetching techniques at a branch, and the extent to which they have been solved is discussed. This leads on to a consideration of the interaction which attempted solutions to these problems have on system software, e.g. paths which are most probable should be fastest, otherwise the speed provided by expensive electronics can be lost.

Although the larger machines are discussed extensively in the undergraduate courses, they do not in themselves constitute the only interest of the department. They are included because it is natural to consider problems such as the provision of large virtual address spaces, in the context in which they first arose, that is, the big machines. Of greater importance, however, is the consideration of matching a machine to its requirements, which has not been given sufficient attention in the past by computer designers (who were not usually computer scientists).

In the treatment of computing systems by the use of mathematical models, too great an emphasis has been placed on the mathematical analysis, whereas a thorough assessment of the basic assumptions on which these models depend would be more scientifically rewarding. It is necessary to stress the need for a more scientific approach, involving rational observation and measurement of the systems that have already been designed.

7. Discussion

Following the representation of section 4, a brief discussion was held. Professor Page commented that much of the teaching related to large, expensive machines. Professor Sumner explained that this was because the problems in which he was interested only occurred in the larger machines. Professor Seitz suggested that these problems were caused by the big machines and that although they were intellectually challenging, they were not necessarily worthwhile. Professor Sumner felt that problems in small machines were due to economic constraints, but in large machines the constraints were of a physical nature and hence more interesting. Professor Wheeler added that today's large machines were the small machines of the future. Professor Seitz expressed his concern that graduates might be biased towards the design of more big, bad machines, to which Professor Sumner replied that he hoped they would build better ones. The remainder of the discussion took place after the presentation of Section 6, when the subject of big machines was again by Professor Horning, who agreed that interesting problems first arose with large computers, and wondered if any attempt was made to convey the idea of a correct size for a machine. Professor Sumner hoped that this emerged from the teaching at Manchester, and pointed out that features once restricted to big machines are now permeating down to the smaller models. Professor Michaelson surmised that manufacturers' efforts were concentrated on the arithmetic unit rather than on control sequencing because the former was easier, with which the speaker agreed, adding that the arithmetic unit is self-contained and that its speed is a marketable feature.

The suitability of the design of Burroughs machines as a subject for study was raised by Professor Suchard, and Professor Sumner commented that this was difficult because of the lack of information available, but this was countered by Professor Barton, who stated that the detailed information on the organisation of Burroughs machines, including a great deal on storage management, would soon be published. He also reported that Wichmann had discovered that compact code, and not ease of compiling, was the main design aim of the Burroughs machines. Their lack of index registers had been initially criticised, but a subsequent comparison of the B5500 with the IBM 7090 favoured the B5500. Professor Randell recalled that it was originally believed that the KDF9 would considerably outperform the B5000 (both machines having an equivalent memory cycle time), and from a limit consideration of inner loops of coding this may have been true. However, a more appropriate viewpoint reverses this opinion. Our ideas as to the correct viewpoint to take are continually changing.

In reply to Professor Seitz, who enquired about the eventual employment of the graduated students, Professor Sumner stated that the majority work in the software field, some go into the electronics industries (for instance the GPO). Of those who take up a career in software, most find employment with computer manufacturers, and this mainly involves system programming. As to their success, though only short-term information is as yet available, they do appear to progress somewhat faster than average. The need to justify the narrow range of both the subjects required for admission to the course and the topics covered by the course, in particular the emphasis on electronic details, was pointed out by Professor Page. The speaker thought that, although an absolute justification could not be given, the course did provide very adequate training for the student in later lift in the computing industry. He felt that the knowledge of hardware acquired was very beneficial, despite the fact that most students eventually specialise in software. The mathematical section of the course, which has not been described here, provides the basic mathematical tools in, for example, statistics. Even so, computer manufacturers, indicate that they would prefer less abstract training in favour of more business data processing.

UNDERGRADUATED TIMETABLE

	ELECTRONIC ENGINEERING	COMPUTER SYSTEM DESIGN	TUTORIAL	PROGRAMMING	APPLICATIONS	NUMERICAL ANALYSIS	MATHEMATICS
1st YEAR	Basic Electricity Magnetism & Electronics 72	Computer System Logical Design 48	24	Programming and Numerical Analysis 48			Analysis & Algebra 96
1st YEAR	Laboratory 80			Programming Practical 80			Examples 24
2nd YEAR	Electronic Circuit Engineering 72	Memory Systems Logic Circuits in a System 48	24	Programming 48	Applications 48	Numerical Analysis 48	Mathematical Methods 48
2nd YEAR	Laboratory 100			Programming Practical 100			Examples 24
3rd YEAR	Computer System Engineering 24			Computing Techniques & Operating Systems 24			
3rd YEAR	Advanced Electronics 48*; Hybrid Computers & Control Systems 48*	Design of Large Computer Systems 48		Systems & Programming 48*	Business Applications 48*	Special Systems 48*	Numerical Analysis 48*
	4 AFTERNOONS PER WEEK PROJECT						

Fig. 1

* indicates optional course

The duration in hours of each course is noted