COMPUTER TECHNOLOGY AT THE UNIVERSITY COLLEGE OF SWANSEA

D. Aspinall

Rapporteurs: Dr. B. G. Cox
Mr. L. B. Wilson

This talk will be in three parts:

1. The general background to the course;
2. The curriculum that has been developed;
3. Some comments on how some of these courses meet the needs in computer design.

1. Introduction

The University College of Swansea, a constituent College of the University of Wales, consists of some 3,400 students in faculties of Arts, Economics and Social Studies, Science, and Applied Science. The computing facilities of the College have outgrown a modest IBM 1620 installation to the point where an ICL 1905E is reaching saturation. The service is used mainly by the Applied Science faculty, but there is also extensive use made of it by all faculties. Around this service has grown a Department of Computer Science under the direction of Professor David Cooper. Mr. Allan Gilmour, formerly of ICSL and Baric, will shortly take up his appointment as Director of the Computer Service. There is thus a firm foundation in the use and science of computers. The Electrical and Electronic Engineering Department is one of the major computer users in their work on circuit design, power system distribution systems and process control, and it inherited the IBM 1620 for experimentation purposes when it was retired from the computer service. It was also one of the first users in the United Kingdom of a PDP-8 computer which is connected to an analogue computer to provide a hybrid system as a vehicle for process control research. The department developed a hybrid language HELP to assist in this work. The research in the department on the design of electronic instruments developed expertise in the use of digital circuit techniques and an awareness that a whole new body of knowledge was developing on top of the basic Electrical and Electronic Engineering.

Professor William Gosling, Head of the Department, realised that this material could not be adequately covered within the framework of the Electrical Engineering Degree Course, and began a series of discussions with Professor Cooper to see if the engineering design and use aspects could not be combined with the Computer Science aspects to constitute a new degree course of Computer Technology.

There was evidence at the University of Manchester that a course of this broad content offered by its Computer Science Department was successful academically, and of use to industry. To collect more evidence and opinion, Professor Gosling wrote to major manufacturers and users of computers for comments about the structure and value of such a course. Their response was uniformly enthusiastic. In addition to support for the marriage of Computer Science and Engineering there came, surprisingly, a strong suggestion that the course should include Applied Economics, to make the students aware of the financial problems of launching a new innovation and the general economic problems faced by a high technology-based industry. Fortunately, the College possesses an Economics Department in sympathy with these broad aims, and its Head, Professor Ted Nevin, eagerly joined in the discussions and suggested existing courses which were relevant and offered to develop new courses where appropriate.

It became clear that this new course could not be made up solely of existing courses in all three Departments, and that all its courses would have to be co-ordinated in a way which could not be achieved by the conventional Joint Honours degree schemes. Thus, it was decided to set up an independent Computer Technology Degree to be supervised by a Board of Studies consisting of all academic staff involved. The first duty of this Board was to draw up the complete curriculum of the course, based on existing courses, and to define the syllabus of each new course.

In view of the spontaneous response from industry, it was decided to constitute an Industrial Advisory Group made up of representatives of

Industry, to advise on the curriculum and its future development. This group meets twice a year, and its members are drawn from computer manufacturers and major users of computers. In the academic staff there was strength in the use and science of computers and in the basic circuit and logic level of computer engineering. What was lacking was the knowledge of computer systems, which bridges the gap between the circuit level and the users. I was recruited in October 1970 to bridge this gap; to take over the Chairmanship of the Board of Studies, and to be responsible for the establishment of the basic syllabus. We are beginning to strengthen this area by the recruitment of more staff with the relevant experience.

The first students entered the course in October 1970. There were twenty-one in the first year, and four who entered the second year by transfer from a computer based first year course. These four were sadly lacking in engineering skills and were brought up to the mark by an extra short Summer Course and by special tutorials during the session. The 1971 intake have been offered places conditional on obtaining 3 C's at A level, and present indications are that between twenty and thirty students will enter the course this October.

2. The Curriculum of the Course

The curriculum is summarised in the attached table, figure 1. There are four major components: Computer Engineering, Computer Science, and Mathematics occur in all three years, whilst Economics occurs in the last two years.

In the first year we attempt to give the students a firm foundation of the basic electrical and electronic engineering required at the circuit level. This is achieved by courses 30.02 and 30.03, each of which consists of three units of lectures and one and a half units of practical laboratory class. These are the same courses as are given to Electrical Engineering students in their first year. Course 63.04 gives an introduction to computer system design embracing information representation, Boolean Algebra, elementary logic design, the design of a basic computer, and the concept of the hierarchy of levels in a computer structure. This course will also be attended by students of Computer Science who have no interest in the circuit level of the technology, and it does not contain detailed reference to the electronics. The circuit level is covered in course 63.10 in which the peculiar problems of digital circuits are discussed. These include: amplification, construction and characteristics of simple circuits, effects in the time domain, the effect of stray capacitance, etc. Programming of a computer in its assembler language brings the students face to face with the problems of using an actual computer and a better understanding of its structure. This experience is provided in the course 63.03 and by programming a PDP-11 computer in practical classes.

It is important that the students become confident in the use of a computer, and are aware of its power. They must therefore be taught a high-level language and be shown how languages provide different features. These languages are taught in 63.01 and 63.02, and practical experience is gained through the 1905E computer service. We find they quickly learn to program quite complex problems.

The mathematics courses are the same as those taught to first year Electrical Engineering students, and are intended to provide a firm foundation for the rest of the course. The Statistics is introduced to meet the later needs of the Economics content.

Electrical Engineering figures strongly in the second year. Courses 32.01, 32.05, and 34.02 are all as taught to Electrical Engineering second year students. In each case the course contents have been considered in the light of the broad structure of the Computer Technology course and the emphasis has been changed to suit it where appropriate. Course 32.10 is an extension of 63.10 and considers further the problem of connecting circuits to operate at high speed. The students should also be aware of the required characteristics of a memory device and how such devices are connected to form a complete memory system. This should be taught in a device independent manner and go on to describe how typical modern memory devices are built into a system. This results in the course on Memory Systems 64.01, which is also taken by third year Electrical Engineering students as an example of an advanced electronic system. The course 35.03 is included to give the students an introduction to the concepts of feedback systems and control which will be developed further in the third year.

All the engineering courses are complemented by a laboratory which includes typical second year electrical engineering experiements plus some specially introduced for the computer technology course. These will include evaluation of a logic circuit family, primary and secondary characteristics of a magnetic memory core, magnetic drum recording, and the construction of a simple computer.

The structure of the high level languages used in the first year is considered and the problems of compilation, interpretation, and assembly of such languages are discussed in course 63.05, together with the definition and design of operating systems. Advanced programming concepts such as Data Structures and Program Control are taught in 63.06. Practical experience is obtained by programming the Computer Service 1905E computer and by short projects based on the PDP-11 computer. The mathematics content develops along lines which will be useful to engineers at the circuit level, and also the application level, in process control. The economics course begins with 51.01 in which the economics of a small unit, such as a firm, are explained. It goes on to discuss the problems of world economics in 51.06.

By the third year, all the Computer Engineering courses become distinct from the Electrical Engineering courses. Advanced Digital Electronics, 64.02, though it is taken by third year electrical students as a further example of applied electronics, is designed primarily to serve the needs of the Computer Technology course. It includes a rigorous treatment of modern logic circuit families, the connection of such circuits, their application in digital systems, and goes on to describe methods of analog-to-digital conversion and related interface techniques. The Computer Architecture course, 64.03, discusses the problems in the design of a range of computers, in particular those of specifying the structure to accommodate a large time-shared computer at the top of the range. Features of large machines are considered, in particular the techniques used in the design of arithmetic units and control of such machines. The concepts raised by Professor Bell this morning may result in the approach to this course being changed to give a wider appreciation of computer structure.

The Hybrid and Digital Simulation course, 64.04, reflects the strong research interests of the department in the application of computers in process control, and in analog-to-digital and digital-to-analog conversion. It is expected that all three third year subjects will evolve together as this research interest grows, and 64.03 develops as a bridge between the Electrical Engineering and Computer Science Departments.

The present research interests in Computer Science are reflected in 63.07, Theoretical Machines, whilst 63.08 is included to widen the students' field of vision of the application of computers. The course on mathematical methods is designed mainly to assist course 64.04, whilst its content forms the basis for the application of computers in many areas. Computer application also comes to the fore in the economics courses which introduce the students to some of the mathematical and computer-orientated techniques used in economic analysis, in addition to a further development of broad economic concepts.

During the third year the students will undertake a single project, in the nature of a small research project, at the end of which the student will produce a thesis which will have a significant effect upon his final grade. The project may be in the application of computers, the software of computers, system design, logic level, circuit level, or even device level. This wide range of topics is the only place where options are offered in the three year course.

The course as it stands at the moment presents a unified whole, and operates within the academic resource of the contributing departments. As new staff are recruited and research expertise develops, we will expect the course content to alter to reflect the development of the subject. But we should not lose sight of the main aim; to give the student the ability to improve the performance of computers and exploit them to serve man's needs.

3. Some Comments on Teaching Computer Design

To integrate this discussion with the topic of this seminar, let us look at those courses which teach the design of computers.

Broadly the course aims to train engineers who can hold their own with other branches of engineering. A rough definition of such a person

.is one who can do for tenpence what a moron can do for a pound. The course has a sound base in practical experience and reflects the influence of the course at Manchester.

The first year courses 63.04 and 63.10 start with simple circuits, and develop through "and" and "or" gates to Boolean Algebra, truth tables, and how to build a simple serial adder. We find students wish to understand just how a machine works at this level. By introducing the concepts of memory and flip-flops, a simple serial computer with cathode ray memory, and with a simple set of instructions (looking strangely like the Manchester Mark I) can be described in detail. This year the structure of the PDP-11 will also be covered.

The second year lectures concentrate on electrical engineering with concepts of computer design being covered mainly by laboratory work. This work is based on register to register concepts. A modular system based on microprogramming is planned. Some of the student's time is spent on small projects on the PDP-11, which has appropriate converters, allowing them to build small instruments as illustrating the design of special purpose processors around a general purpose one. They also experiment with different types of memory components and so complement the lectures by determining the primary and secondary characteristics of a memory device. We hope by this they would later be able to ask the appropriate questions if confronted with a new memory device. The work also includes the evaluation of TTL circuits, the speed limitations, and the problems associated with connecting elements together.

The third year deals with the parts of a machine, the design of a high speed arithmetic unit, buffers, slave stores, and similar devices. Students may choose to do major projects on the design of special purpose computers, or on software in a general purpose machine on the PDP-11.

In brief then we attempt to deal with a few concepts in depth rather than too many too shallowly, and endeavour to show how computer design fits into the total computer technology hierarchy.

COMPUTER TECHNOLOGY COURSE AT THE UNIVERSITY COLLEGE OF SWANSEA

Figure 1.

PART I

			Units	
			L	P
COMPUTER ENGINEERING	30.02	Electrical & Electronic Engineering I	3	1½
	30.03	Electrical & Electronic Engineering II	3	1½
	63.10	Computer Circuits	1	3
	63.04	Introduction to Computer Systems	2	
	63.03	Basic Machine Programming	1	
COMPUTER SCIENCE	63.01	Programming Languages I	2	8
	63.02	Programming Languages II	3	
MATHEMATICS	10.08	Mathematics I	3	1
	10.09	Mathematics & Statistics	4	1
ECONOMICS			22	16
		Total	38	

One unit = 12 lectures of 12 hours.

PART II

			Units	
			L	P
COMPUTER ENGINEERING	32.01	Electronic Circuits I	3	10
	32.05	Digital Circuits	1	
	64.01	Memory systems	2	
	35.03	Introduction to Control	1	
	32.10	Interconnections and Transmission lines	1	
	34.02	Communications Systems	2	
COMPUTER SCIENCE	63.05	Systems Programming	4	3
	63.06	Data Structures and Control	3	
MATHEMATICS	10.03	Engineering Mathematics	3	1
ECONOMICS	51.01	Introduction to Micro Economics	2	
	51.06	Principles of Macro Economics	2	
			24	14
		Total	38	

PART III

			Units	
			L	P
COMPUTER ENGINEERING	64.02	Advanced Digital Electronics	2	
	64.03	Computer Architecture	2	
	64.04	Hybrid and Digital Simulation	2	
COMPUTER SCIENCE	63.07	Theoretical Machines	3	3
	63.08	Numerical Analysis	3	
MATHEMATICS	10.28	Mathematical Methods	2	
ECONOMICS	51.02	Industrial Efficiency	2	
	51.07	Econometrics	4	
	51.04	Business Accountancy I	1	
	51.05	Business Accountancy II	2	
	PROJECT			10
			23	13
		Total	36	

Discussion

Professor C. L. Seitz:

I am surprised by your comment that "an engineer can do for tenpence what a moron can do for a pound". Do you mean penny-wise pound foolish, or should we take the remark more seriously?

Professor Aspinall:

We are concerned with the right and wrong way to put circuits together. An engineer will look at a problem, weigh up the alternatives, consider the economics of using existing equipment or, if necessary, designing something new, and will find the cheapest way to produce the required processor.

Seitz:

Is this type of approach unique to engineers or can it be developed by other disciplines?

Aspinall:

It is an approach common to all engineering and is based on experience gained over the years.

Mr. H. C. Lauer:

I am concerned about the difference between the engineer as you define him and the architect as defined this morning by Professor Barton. Is the engineer concerned with say the structure of a bridge, while the architect would consider the bridge within the fabric of, say, a city? Why then the set of economic courses which are not particularly related to the rest of the year's work - why not something like Homeric poetry instead?

Aspinall:

The Computer Architecture course is supposed to tie together the hardware and software parts of the course. We are aiming to turn out a man who can not only go and build a machine, but also understand one he is confronted with, and decide the best way to use it. The inclusion of courses on Economics is in some respects an experiment as both industry and the University felt the need for such courses, which had been given to electrical engineers for some time. Since most of our graduates will end up in industrial concerns we feel they should know how a firm operates and how it fits into the world around it. We try to cover techniques which are generally useful, and if possible show how computers fit in.

Professor D. C. Cooper:

The course was not designed for manufacturers specifically but for the industry in general.

Professor E. S. Page:

I am worried about some aspects of design in general, particularly as it relates to computer system costs. With the cost of components varying so rapidly even within a year, it must surely be impossible to specify a framework for design unless cost limits are set.

Aspinall:

This is appreciated and dealt with artificially in the project by specifying a set budget and a completion date. All the time we emphasise the need to find a globally cheapest way of achieving the result within the constraints applied.

Professor C. G. Bell:

From the description you have given it appears that you do little in the formal theory of optimisation. Is this so?

Aspinall:

We do a little with truth tables, and if there is time, we cover topics like Karnagh maps and minimisation techniques.

Bell:

I would suggest that standard optimising techniques like linear programming could well be useful if there is not room for a proper introduction to optimisation theory.

Aspinall:

Perhaps these will be introduced later, but at present there is not sufficient time to include all we might wish.

Professor B. Randell:

I am worried about the concern to evaluate only on the basis of costs, and I suspect I have a different definition of computer architecture. However, in many areas, for example, the value of relocation hardware, it is difficult to assign costs accurately, yet important decisions are based on the results.

Aspinall:

There are real difficulties in getting estimates from software people of what will give say a 10% improvement to make it worthwhile to do something a different way. While we would need more time to cover all aspects in detail we use the adder as a practical example of how pipe-lining techniques and other improvements can be carried out, and the cost of such improvements.

Randell:

While the hardware people can often make accurate cost estimates, it is often not possible in software areas.

Aspinall:

One should not think that a specific architecture is agreed overnight. Instead it is defined over a long period of time with give and take from both hardware and software sides of the fence. There is usually evidence of the cost of production of the required hardware facility, but it is difficult to assess its effect on software costs.

Professor J. J. Horning:

I do not agree that detailed knowledge at this level is necessary, Is it not the same as saying that to be able to drive a car one must know in detail how a spark-plug works?

Aspinall:

I cannot accept this. There is no way to discuss costs without knowledge of the detail of the components. This may be done by looking at the evolution of computer systems by means of examples.

Horning:

It sounded as though this was used to mean architecture.

Aspinall:

Regrettably there is not enough time to have two separate courses. We would like to spend more time discussing computer structure but we do make an attempt to relate software and hardware problems. I must repeat that I have concentrated, in my lecture, on that part of the course to do with the design of the computer hardware.

Professor E. S. Page:

Any example chosen for analysis could be a target for the criticism of too much attention to detail.

Aspinall:

This is true. While I have tended to concentrate on the adder, several other examples are covered during the course.

Professor C. L. Seitz:

Have we now got you in a corner - are you saying that all must know how an adder works?

Aspinall:

We consider that every student on our course should understand circuits so that they can assess a device for use in computing and assemble such devices into a working system.

Professor D. C. Gilles:

Is it not better to have a few theories rather than several examples?

Aspinall:

I agree. The few theories which exist and are known to be useful are taught.

Professor D. C. Cooper:

The course on Theoretical Machines (63.07) should tidy up some of these concepts.

Mr. H. C. Lauer:

If you got tired of teaching adders, what would you choose as a replacement?

Aspinall:

We are not at that point yet.

Professor C. G. Bell:

Multipliers would serve.

Aspinall:

We take students to the point where they can sit down and design an adder or a multiplier - and make it work. We try to make them understand that this is not the be all and end all of computer design but we hope that by knowing how to design a part of a computer system they can go on to apply this to problems of design elsewhere in the system.