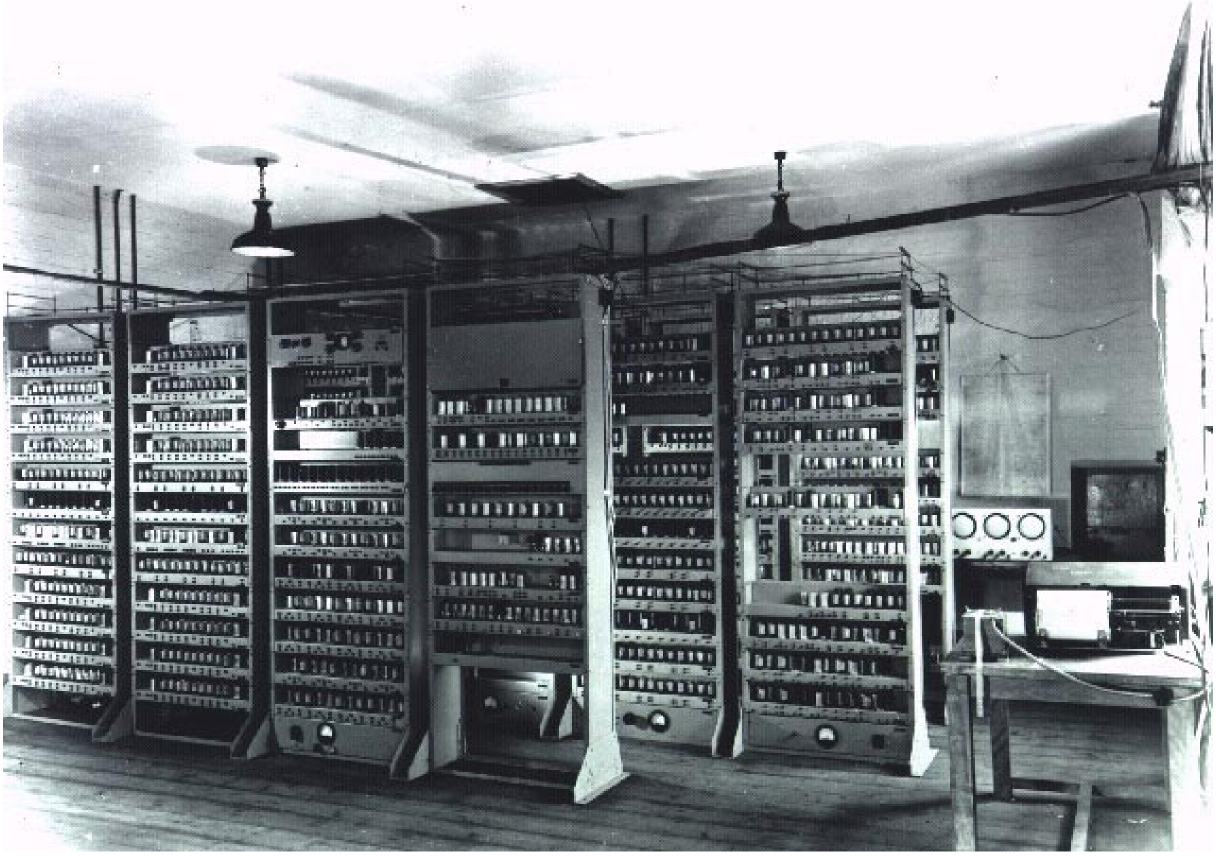
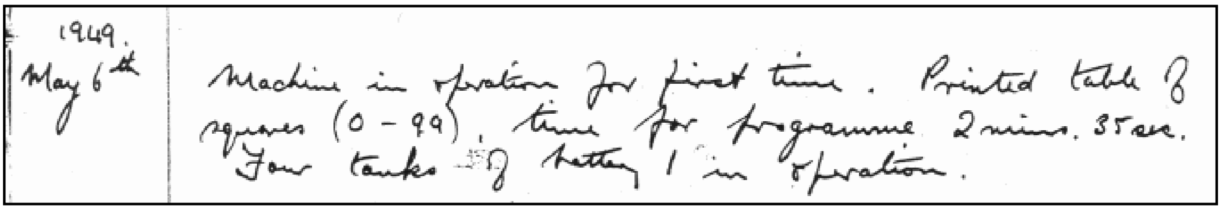
EDSAC 99

15-16 April 1999

UNIVERSITY OF
CAMBRIDGE





1949.
May 6th

Machine in operation for first time. Printed table of squares (0 - 99), time for programme 2 mins. 35 sec. Four tanks of battery 1 in operation.

Computer Laboratory
New Museums Site
Pembroke Street
Cambridge CB2 3QG

http://www.cl.cam.ac.uk/

EDSAC 99

15-16 April 1999

Peter Robinson & Karen Spärck Jones

Second edition with minor corrections, 6 May 1999.

Preface

The EDSAC ran its first program on 6 May 1949, computing a table of squares. The Computer Laboratory is celebrating the fiftieth anniversary of this historic event, when the full-scale electronic, stored program, digital computer built by Maurice Wilkes and his team came into operation. The work done by these pioneering forty-niners marked a major step in the development of modern computers.

Members of the Mathematical Laboratory in May 1949

Top row, from left:
D. Willis, J. Stanley, L. Foreman, G. Stevens, S. Barton, P. Farmer, P. Chamberlain.
Middle row, from left:
H. Smith, C. Mumford, H. Pye, A. Thomas, E. McKee, J. Steel.
Bottom row, from left:
R. Bonham-Carter, E. Mutch, W. Renwick, M. Wilkes, J. Bennett,
D. Wheeler, B. Worsley.

"Finally, and rather suddenly, on 6 May 1949, the machine read in a program tape for computing a table of squares and printed the results. David Wheeler immediately set about writing a program for computing prime numbers and a day or two later this program had also run."
[M.V. Wilkes, Memoirs of a Computer Pioneer, MIT Press, 1985, p142.]

Introduction

The EDSAC 99 celebration is designed to capture the work and life of the Laboratory at the time that EDSAC was built and came into use for research and then teaching, and to follow the development of the Laboratory in the fifty years since 1949 through its research, teaching and the provision of a computing service. The meeting features talks by the pioneers themselves, and by later members of the Laboratory; a show of historic exhibits focussing on the EDSAC, in its context of the Laboratory's earlier calculating machines and successor machines to the present day; some EDSAC simulators; demonstrations of current research; and an exhibition on the Computing Service.

PR & KIBSJ
Cambridge
April 1999

We are very grateful to the following for their sponsorship of EDSAC 99:

- ARM Limited
- AT&T Laboratories Cambridge
- Cambridge Crystallographic Data Centre
- Marks and Spencer plc
- Microsoft Research Limited, Cambridge

Contents

Preface 1

Introduction 3

Contents 5

Programme 7

Thursday 15 April 7

Friday 16 April 8

Abstracts of talks 9

Robin Milner: *Introduction – a new discipline* 9

Maurice Wilkes: *EDSAC 1 – getting it all going* 10

David Wheeler: *EDSAC 2* 11

The Engineers: *Incidents and anecdotes* 12

Roger Needham: *The Titan years* 14

David Hartley: *Serving the University* 15

Andy Hopper: *Rings and things* 16

Michael Gordon: *Proving computers correct* 17

Frank King: *Teaching it straight* 18

Stewart Lang: *Displaying dimensions* 19

John Daugman: *Eyeing the user* 20

Ian Leslie: *What about the future?* 21

Karen Spärck Jones: *Conclusion – getting there* 22

Historic objects exhibition 23

EDSAC simulators 33

Current research 35

Self-timed logic (Simon Moore & Peter Robinson) 35

Autostereo 3D display (Neil Dodgson & Stewart Lang) 36

Next generation workstations (Ian Pratt & Austin Donelly) 37

Warren home area network (David Greaves & Daniel Gordon) 38

The Nemesis operating system (Steven Hand & Paul Menage) 39

The active house (Jean Bacon, Andrew McNeil & Alexis Hombrecher) 40

Network control & management (Ian Leslie & Richard Mortier) 41

Proving protocols correct (Larry Paulson & Gianpaolo Bella) 42

XIsabelle – supporting proof (Katherine Eastaughffe) 43

Floating point verification (John Harrison & Myra van Inwegen) 44

Modelling interactive systems (Philippa Gardner) 45

Natural language processing (Steve Pulman) 46

Iris recognition (John Daugman) 47

Modelling and animation (Neil Dodgson & Peter Robinson) 48

Video user interfaces (Peter Robinson & Richard Watts) 49

Computing Service exhibition 51

New Computer Laboratory Building plans 55

Extracts from the 1949 conference 57

A brief informal history of the Computer Laboratory 65

Acknowledgements 75

Postscript – the Bun Shop 76

Programme

Thursday 15 April

11:00-1:30	Historic objects exhibition	Common Room
	EDSAC simulators	Discussion Room
	Current research – demonstrations and displays	Cockcroft 4
	Computing Service exhibition	Austin staircase
	New Computer Laboratory Building plans	TP4
2:00-5:30	Talks	Babbage Lecture Theatre
2:00	Professor Robin Milner *Introduction – a new discipline*	
2:15	Professor Maurice Wilkes *EDSAC 1 – getting it all going*	
3:00	Professor David Wheeler *EDSAC 2*	
3:30-4:00	Break	
4:00	Computer Laboratory Engineers *Incidents and anecdotes*	
4:30	Professor Roger Needham *The Titan years*	
5:00	Dr David Hartley *Serving the University*	
5:45 & 6:00	The EDSAC film	Babbage Lecture Theatre
5:30-6:15	Historic objects exhibition	Common Room
	EDSAC simulators	Discussion Room
	New Computer Laboratory Building plans	TP4
6:15-7:15	Reception	University Combination Room
7:30-	Dinner	St John’s College Hall
	Speaker – Professor Sir Peter Swinnerton-Dyer	

Friday 16 April

9:00-12:30	Talks	Babbage Lecture Theatre
9:00	Professor Andy Hopper *Rings and things*	
9:30	Professor Michael Gordon *Proving computers correct*	
10:00	Dr Frank King *Teaching it straight*	
10:30-11:00	Break	
11:00	Dr Stewart Lang *Displaying dimensions*	
11:30	Dr John Daugman *Eyeing the user*	
12:00	Professor Ian Leslie *What about the future?*	
12:15	Dr Karen Spärck Jones *Conclusion – getting there*	
12:30-3:00	Historic objects exhibition	Common Room
	EDSAC simulators	Discussion Room
	Current research – demonstrations and displays	Cockcroft 4
	Computing Service exhibition	Austin staircase
	New Computer Laboratory Building plans	TP4

Abstracts of talks

Robin Milner: *Introduction – a new discipline*

There is no doubt that in May 1949 a new form of human activity began, which is already more significant than telephone and television, and will soon outstrip printing. These are all innovations which put a new engineering triumph in the hands of ordinary people, and they are all things whose impact on our lives could not be fully perceived at the start. What was achieved in the (then) Mathematical Laboratory at Cambridge was a marriage of all the subtly-related ideas about program, memory, subroutine, self-manipulation etc. with the right technology, to provide ordinary people (at least, ordinary scientists) with something so reliable that not only could they admire the ideas intellectually, but they could use them in a commonplace manner. This was the miracle, for so it now seems.

The outcome of this new human practice was twofold; first, a computing service to manage the practice; second, a new intellectual discipline arising from the practice. (These are stated in no particular order!) In other words, the University's teaching and research into computer science and the University Computing Service were born as the same infant, or as Siamese twins, in Cambridge in 1949.

This infant is now two 50-year-old adults, and in EDSAC 99 we are going to hear what they have been doing all their lives. If you follow the history of the whole Laboratory, you are struck by the way the two activities have been interwoven. Actually, it soon became three activities because the academic part developed its teaching in tandem with its research. On the service side you can see how one machine after another, developed from research, became the computing servant; on the academic side you can see how the practice of computing led to one research strand after another; new programming languages, initiatives in communication, computer-aided design, formal foundations, and recently a flurry of new things. Teaching, as it should, has always developed in order to transmit whatever shows signs of lasting more than a few years.

In these two days we are lucky to be able to hear from so many who took part in this exciting development. The greatest luck of all is to have with us Maurice Wilkes and David Wheeler, who began it all, to give us a first-hand report of the beginnings. The concerns of those who follow can never be the same as those who create a breakthrough, because that very event has changed our thinking. But we hope to gain some impression of how they thought and felt, when they gave computing practice to the world at large. It is a very great privilege for me to chair a session in which these two pioneers can both take part.

Robin Milner went on an EDSAC course in 1956, but only joined the computing profession (Ferranti) in 1963. After spells at the City University, University College Swansea and the AI Laboratory at Stanford University he joined the teaching staff at Edinburgh University in 1973. He came to Cambridge in 1995, and took over as Head of the Computer Laboratory on 1 Jan 1996. He has mainly worked in semantic theories of computation, particularly of interactive systems; also in machine-assisted reasoning and programming language design.

Maurice Wilkes: *EDSAC 1 – getting it all going*

It is not easy to get over to people today what Cambridge was like in September 1945 when I was released from war service. There was universal excitement in the air and an infectious enthusiasm to re-establish peacetime values after the madness of war. At all levels in the University, there was encouragement for those who knew what they wanted to do. This included a sympathetic attitude to funding.

My remit was to get the Maths Lab going as an academic department and to initiate a research programme. I first had to find out about the developments in computing machinery that had taken place in the United States during the latter part of the war. My own war work had been largely concerned with radar and the sort of electronics that went with it.

Fortunately, there were people on hand from whom I could learn. My problem was to be sure that I had learned everything that was relevant. It was attending the famous course of lectures in Philadelphia that gave me confidence. When I walked up the gang plank of the RMS Queen Mary in September 1946 on my way home, I felt that I knew what had been done and had some idea of what I could perhaps attempt.

During the voyage, I sorted out my ideas and made a few rough notes. Back in Cambridge, an able team began to gather around me. They brought their own skills and ideas to bear, and it was their efforts that made the EDSAC a reality. They gave the Maths Lab its own special atmosphere and made it an exciting place to work.

Maurice Wilkes went up to Cambridge University in 1931 and studied mathematical physics and other subjects. In 1934 he became a graduate student in the Cavendish Laboratory, doing experimental research on the propagation of radio waves in the ionosphere. This led to an interest in tidal motion in the atmosphere and his first book was on this subject. It also led to an interest in computing methods and, when he returned to Cambridge in 1945 after war service, he became head of the Computer Laboratory, then called the Mathematical Laboratory.

In the summer of 1946, Wilkes attended the famous Moore School lectures on electronic computers in Philadelphia. On his return, he set about building the EDSAC, which began to work in May 1949. In 1951, he published jointly with two colleagues, David Wheeler and Stanley Gill, the first book to appear on computer programming. At this time, he put forward his proposals for microprogramming, a system which later became adopted widely in the industry. In 1965, he published the first paper on cache memories, followed later by a book on time-sharing.

In 1974, it appeared to Wilkes that the time had come when local area networks based on traditional telecommunication technology might profitable be replaced by networks of much wider bandwidth based on computer technology. The design study for what became known as the Cambridge Ring was published in 1975. The Cambridge Model Distributed System, a pioneering client-server system, described by Wilkes and Needham in 1980, was based on this ring.

Since 1980 Wilkes has worked in industry, first with DEC in Massachusetts and now with AT&T Laboratories, Cambridge, England, where he is a staff consultant.

David Wheeler: *EDSAC 2*

This was the second computer built in the Mathematical Laboratory. Design work started with a grant from the Nuffield Foundation in 1953, exploiting the knowledge built up from the operation of EDSAC 1. EDSAC 1 was also used in the design and making of the wiring schedules.

EDSAC 2 was a microprogrammed machine. The microprogram was held in the wiring of a control matrix of 1024 cores of 13mm ferrite cores. This made the concept eminently practical. EDSAC 2 was very 'user friendly', since the usual anomalies were coped with in the microprogram and ROM.

To ease maintenance, it had a plugged bit slice design for the arithmetic unit and index registers. The store used ferrite cores and we initiated their production at Mullards. Memory size ferrite cores (2mm) were used in a ROM in which each core held four bits but used the same amplifiers and drives as the ordinary memory. The 768 words held all the commonly used routines such as assembler, print, trigonometric functions etc.

The input and output were by paper tape. Input used laboratory designed paper tape readers which operated at 1000 characters per second and could stop instantly. The output was considerably slower initially. However in the course of EDSAC 2's life many output devices were added including a line printer, curve plotter, photographic imager, and two high speed punches.

Magnetic tapes and a store extension of 16K words added to the machine's power for the bigger jobs, and enabled an Autocode to be used.

The EDSAC 2 lasted about a decade and had many hundreds of users and programmers. It caused most departments to become computer conscious, which stood them in good stead for things to come. In my opinion it was the most successful of the laboratory computers. It utilised the experience gained in running EDSAC, it did not have wearisome compatibility requirements, it matched the current user needs admirably, and gave the greatest step forward in performance of the laboratory computers.

Two Nobel prizes arose from work done on the EDSAC 2, but more importantly the University became computer aware and originated calculation methods in many fields.

David Wheeler joined the Mathematical Laboratory in June 1948 as a research student. Wrote the first programs for the EDSAC. Awarded the PhD in 1951 (the first in Computer Science). Went to the University of Illinois 51-53. From 1953 worked on the design of EDSAC 2 and has been contributing to Laboratory projects since then. Retired in 1994.

The Engineers: *Incidents and anecdotes*

In 1947, **Maurice Wilkes** was joined at the Mathematical Laboratory by **Eric Mutch** from TRE Malvern and **Philip Farmer**. Also came **Bill Renwick**, a design engineer who had worked on radar for the Royal Navy. These four then began to recruit a team of engineers to built EDSAC 1. They came from a variety of backgrounds with differing skills. As far as can be remembered they were:

> **Gordon Stevens** – an instrument maker from Unicam Instruments.
>
> **Sid Barton** and **Peter Chamberlain** – both straight from the RAF after the war had finished.
>
> **Roy Piggott** – straight from school.
>
> **Bill Lenaerts** – seconded from Joe Lyons.

They were joined later the next year by:

> **Dick Kimpton** – straight from school.

Then a bit later still by:

> **Len Foreman** and **Ivor Reynolds** – also from school.

From the early 1950s many more engineers were recruited to work on EDSAC 2.

Components for EDSAC were purchased either through ordinary trade channels or from a Government store that we were allowed to use. The valves were a gift from the Ministry of Supply of (unused) surplus stock. However, much local adaptation and ingenuity was required. Gordon Stevens built the first paper-tape reader. Paper-tape punches and printers were obtained, but had to be modified to work on EDSAC.

All the engineers wore white coats, a necessary precaution. Fault finding was very much an hit and miss affair in those early days, with the only diagnostic instruments being an Avometer and a fairly primitive oscilloscope. Valves were changed sequentially until the faulty one was located.

There must be a lot of stories to retell from those early pioneering days, and hopefully we will be able to recall some of them, from those who are left.

Gordon Stevens came from Unicam Instrument Co. as the Lab's first Scientific Instrument Maker in 1947. In 1949 he became Senior Assistant looking after accounts and generally anything that needed attention. Retired in 1982.

Richard Kimpton started at the Mathematical Laboratory straight from school in September 1948 as a lab assistant. Worked on EDSAC 1 and after returning from National Service in 1954, did all the backwiring on EDSAC 2 in the next 12 months. Constructed peripherals for Titan and the prototype page for CAP. Retired in 1998.

Herbert Norris joined the Lab from Engineering in 1950 as an Instrument Maker to assist Gordon Stevens. Left in 1965 to take up a teaching post at the Manor School. Now retired.

Ivor Reynolds started straight from school in 1950 as a junior lab assistant. He was involved in building several prototype experimental circuits. He left to work for De Havilland Propellors in 1958/59, but returned in 1960 for a year to design Chassis 3 for EDSAC 2. Then went to British Aerospace. Retired in July 1993.

Vic Claydon joined Maths Lab from Engineering in 1951 as an Instrument Maker. Worked on both EDSAC 1 and 2. Then headed the team of Mechanical Engineers working on Titan, maintaining all the peripheral equipment. Was involved in the early days of the Hardware Maintenance Service. Retired in 1982.

Ken Cox joined the Lab in December 1958 from Pye, to work as a Maintenance Engineer on EDSAC 2. Moved to Titan in 1968. Set up the Hardware Maintenance Service in 1979 and on Sid Barton's retirement became Chief Engineer in 1982. Retired in 1993.

Peter Bennett joined from Pye in March 1959 as a Maintenance Engineer for EDSAC 2. Trained to maintain the DEC PDP-7. Worked on Titan and then CAP and the Cambridge Data Ring. Retired in 1989.

John Loker came from Pye in April 1959 to work on peripherals in the Tape Preparation Room. A year later joined the team of engineers on EDSAC 2. Worked on the EDSAC 2 Lineprinter with Norman Unwin. In 1966 helped to design and build the Multiplexer for Teletypes on Titan. Retired in 1998.

Roy Bayley joined the Lab in 1961/62 from Marshalls to help with the commissioning of Titan, as Sid Barton's deputy. Left in 1973 to work for the MRC Human Genetics Group in Edinburgh. Retired in the Autumn of 1995.

David Prince joined the Lab in June 1963 from Jodrell Bank to work on the Tunnel Diode Slave Store for Titan. A year later joined the team of engineers first commissioning and then maintaining Titan. Worked on CAP, and then in 1979 started the Hardware Maintenance Service with Ken Cox. Appointed Chief Engineer in 1993.

Roger Needham: *The Titan years*

The Lab scene in the sixties and on was hugely influenced by the Titan project. It was the last time we had a major part in building the machine that would provide the University's principal service, which was done because of poverty. However the Titan had innovative hardware, had an operating system that in structure and concepts was way ahead, and led in an indirect way to advances in understanding of programming languages. While the system was being developed Maurice Wilkes became aware of activities at MIT which led to the CTSS – the first time-sharing system of its type. He came back determined that Cambridge should be in that game, and the Titan project, both hardware and software, was wrenched onto a new course. In the event a system resulted that was far from being a CTSS copy and had some good features that were rarely found much later. The multiple-access system was thrown open to the university public in early 1967 – and this would almost certainly not have happened had we been richer in 1960.

In 1965 a substantial grant was obtained from the Science Research Council to exploit the facilities that were then clearly predictable. It was entitled "Research in Computer Science" and left the judgement of what actually to do entirely to Wilkes – a contrast with the Swindon bureaucracy of today with its deliverables, beneficiaries, progress charts, milestones, millstones, and what have you (you have to go into industry to avoid that now – *experto crede*). As a result of recruiting Charles Lang, one of the things done was to go in for research on mechanical CAD. This activity lasted fifteen years, and out of it came numerous local companies and the CAD Centre, at first a government project operated under contract by ICL. The impetus for setting the CAD Centre up was specifically the Lab's operating software; the Centre still flourishes as an independent company and is tangible evidence of the successes of the sixties.

Roger Needham used the EDSAC when taking the Diploma in Numerical Analysis and Automatic Computing in 1956-7. Became a research student, then employee in the Mathematical Laboratory in 1962, where he continued to work, mainly in the systems area, until 1998. Head of Department 1980-95. Now Managing Director of Microsoft Research Limited in Cambridge.

David Hartley: ***Serving the University***

In 1937 the remit of the Mathematical Laboratory included the requirement "to provide a computing service for general use, and to be a centre for the development of computational techniques in the University." But computers as we now know them did not exist, so a good deal of R & D was needed before this remit could be satisfied. It might be said that the Computing Service was the horse to be followed, very closely, by the cart of Computer Science.

In 1969 it was stated that, "over the last twenty-five years the Laboratory has become a service department comparable only with the University Library." Computing as an academic subject and computing as a University-wide service had gone along hand-in-glove for a good many years, the one barely distinguishable from the other. But the distinction became more marked and was formally realised with a major reorganisation in 1970.

Between 1937 and 1970, computers were difficult to make, difficult to keep running and difficult to use. Since 1970, everything has become progressively easier until today, where every academic has at least one computer on his/her desk, and the Computing Service has changed greatly as a result.

I will illustrate some of the important features of the development of the Computing Service that characterise these changes.

David Hartley took the Diploma in Numerical Analysis and Automatic Computing in 1958-59, then research student in programming languages and compilers – developed EDSAC 2 Autocode. Variety of research posts, mainly working on CPL and the Titan operating system; University Lecturer in 1967. Director of the University Computing Service from the re-organisation in 1970 until 1994. Left to become Chief Executive of UKERNA, a company developing and operating the JANET network. Since 1997, Executive Director of the Cambridge Crystallographic Data Centre; also currently Deputy President of the British Computer Society.

Andy Hopper: ***Rings and things***

Work on high-speed communications begun in 1974 with the design and construction of the Cambridge Ring. This provided 10 Mbps of shared bandwidth and ultimately up to 100 devices were interconnected on three rings. An integrated version was begun in 1977, together with the associated CAD project. This was a period of expansion and co-operative effort which included faculty, engineers, and students in what was still essentially a small laboratory.

Faster rings were constructed in the 1980s and operated at speeds of up to 100 Mbps for the Cambridge Fast Ring (CFR), and up to 1 GBps for the Cambridge Backbone Ring (CBR). The latter ran round Cambridge and interconnected local CFR clusters. Considerable experience of VLSI design continued to be acquired (not always positive), which in turn gave a stimulus to the formal verification research going on in the laboratory.

The plentiful bandwidth provided by these pioneering communications architectures made possible the construction of many experimental applications. These ranged from early client/server systems at the end of the 70s (Cambridge Model Distributed System) to networked multimedia systems at the end of the 80s (Pandora).

Some of the devices attached to these networks are noteworthy. An early wire wrap machine was essentially a stateless peripheral. The first laser printer gave up the ghost after almost a quarter million copies. The ring telephone systems anticipated correctly the quality of audio on the internet. The video applications could distort people like a crazy mirror.

The design and use of communications systems has been an important part of life in the laboratory for many years and has helped pave the way for the internet today. It has also been tremendous fun.

Andy Hopper received the BSc degree from the Univerity of Wales in 1974 and the PhD degree from the University of Cambridge in 1978. He was elected a Fellow of the Royal Academy of Engineering in 1996. He is Professor of Communications Engineering at the Cambridge University Engineering Department. He is Managing Director of AT&T Laboratories Cambridge (formerly the Olivetti & Oracle Research Laboratory), and a Founding Director of Virata Ltd, Telemedia Systems Ltd, Adaptive Broadband Ltd and Acorn Computer Group plc.

Michael Gordon: ***Proving computers correct***

The Computer Laboratory was an early pioneer of the “formal verification” of hardware. Formal verification differs from traditional simulation-based verification in using automated reasoning software to construct mathematical correctness proofs. These proofs establish correct functioning for all possible sequences of inputs, not just for a finite number of simulated test cases.

One of the first formal verifications of a complete computer design was done at Cambridge. The machine was trivial by today’s standards (twenty six micro instruction, eight machine instructions, six registers and controlled by a four way switch on the front panel), however it became a standard verification benchmark. The proof was performed by Jeff Joyce, a researcher visiting from the University of Calgary. He subsequently made a single-chip implementation of this computer at Xerox PARC and then came to Cambridge as a graduate student.

A major project at Cambridge in the 1980s was the verification of the VIPER chip, designed at the Royal Signals and Radar Establishment. This was probably the first successful application of formal verification to a real design. The research was done by Avra Cohn and Wai Wong at Cambridge and laid the foundation for later more advanced work around the world. The VIPER project also raised many issues about what was actually achievable by formal verification, and provoked much, sometimes heated, discussion.

Today, in the late 1990s, processors are so complex that the examples studied in the 1980s look ludicrously simple. The formal verification of complete processor designs is now considered intractable and work focusses on fragments, like pipelines. However, this may only be a short-lived phase and there are signs that whole processor formal verification soon will again become an active research area.

Mike Gordon learned numerical analysis from Maurice Wilkes when studying mathematics at Cambridge. Worked for a PhD from the Department of Machine Intelligence in Edinburgh and then returned to Cambridge for the Diploma in Linguistics, followed by first job as research assistant to John McCarthy at Stanford. Returned to Edinburgh to work as a research assistant to Robin Milner, then back to Cambridge for good as Lecturer, then Reader, then Professor.

Frank King: ***Teaching it straight***

The 1953-54 Diploma Course in Numerical Analysis and Automatic Automatic Computing was the first formal course leading to a university qualification in computing anywhere in the world. This talk charts the progress of both the in-house teaching of computing and the extra-departmental teaching since those early days.

Over the years the in-house teaching has witnessed the successive introduction of the one- two- and now three-year Computer Science Tripos, the term "computer science" itself reflecting the development of the subject. Some early projects would also now be more suitable as introductory exercises. Associated with the changes in content have been changes in the assessment methods.

Until the late 1960s the principal extra-departmental teaching consisted of students from other departments sitting in on selected Diploma lectures. A major development was the introduction of television by David Hartley in 1968. Other notable milestones include the introduction of courses for Arts students, and the incredible persistence but eventual demise of FORTRAN.

There has been a long-term trend away from teaching programming to the teaching of applications. Almost everyone now uses some kind of word-processor and even mathematicians have appreciated that spreadsheets are not just tools for accountants.

Now a Lecturer in the Computer Laboratory, Frank King was supervised in the Mathematical Laboratory as an undergraduate in the early 1960s and became a serious user when a research student. He was appointed a University Demonstrator by Wilkes in 1969 and charged with overseeing extra-Departmental teaching. This teaching has changed many times but he is still doing it.

Stewart Lang: ***Displaying dimensions***

“A picture is worth a thousand words”, or so the traditional saying goes.

Where the picture will lead us in the next 50 years is unwittingly a matter of daily interest to all computer users. The vast majority of computer input and output today is through computer displays. Yet, the first interactive displays of the 1965 era in Cambridge were hardly user friendly and were clearly items of major capital expenditure involving racks of electronics and great leaps of faith by the funding bodies. The input/output model of the computer from the 1940s to the 1970s was one of paper – paper tape, paper cards and teletype or line printer listings. Within a decade the whole paradigm for users had changed forever, and arguably for the better.

Today interactive graphics is commonplace for computer output, and a major component of most user input systems: from desktops to handhelds, from games to number crunching, from the sublime to the ridiculous, it is now the computer display that is the predominant medium for interaction. The input/output model is one of moving and transitory images in phosphor, liquid crystal, or light emitting diodes. The Lab has been seen all of these changes, typically from the viewpoint of the software and system developer; at times, also, from the viewpoint of the display developer.

This talk will look back at some of the displays in the history of the Lab. The talk will also look forward briefly to the next generation of displays available to the Lab and its user community in Cambridge. Looking back: from the PDP-7 and Type 340 cathode ray tube display of 1965 to the colour desktop displays of today. Looking forward: from the smallest postage stamp size display to the truly wall-sized day-glo billboard, and from the luminescent plastic pinhead to the fully three-dimensional display.

The change from hard copy to soft copy has been an outstanding phenomenon of the last 50 years of computing. But, the next 50 years offer even greater changes in displays than the last 50 years, and even greater changes in soft copy paradigms to go with those new displays.

What then will a picture be?

Stewart Lang started with computer graphics in 1970 on the PDP-7/340 linked to Titan, during the Diploma in Computer Science. Became a research student and used the PDP-11/Vector General graphics system extensively in his PhD research in the Lab. Left the Lab in 1975 to develop system software for microcomputers. Co-founded Micro Focus Group plc in 1976 (now MERANT plc), which developed COBOL compilers for microcomputers and moved with Micro Focus to Palo Alto. Returned to the Lab as a visitor in 1991 to investigate autostereoscopic graphics displays. Co-founded ASD Systems Limited in 1993 to commercialise autostereo technology, and is now the company’s Chief Executive.

John Daugman: ***Eyeing the user***

Iris recognition illustrates work in computer vision, pattern recognition, and the man-machine interface. The purpose is real-time, high confidence recognition of a person's identity by mathematical analysis of the random patterns that are visible within the iris of an eye from some distance. Because the iris is a protected internal organ whose random texture is stable throughout life, it can serve as a kind of living password that one need not remember but one always carries along. Recognition decisions are made with confidence levels high enough to support rapid exhaustive searches through national-sized databases.

Algorithms I have developed in Cambridge are today the basis for all iris recognition systems worldwide. In America and Japan, the main uses are physical entry control and various Government programmes. In Britain, The Nationwide Building Society introduced iris recognition into its cash dispensing machines (in lieu of PIN numbers) in 1998. Beyond High Street banking, iris recognition is forecast to play a role in a wide range of other applications wherever a person's identity must be established or confirmed. These include passport control, electronic commerce, benefits payment, building entry, access to privileged information, driving licenses, forensic and correctional applications, computer login, or any transaction in which personal identification currently relies on keys, cards, documents, passwords or PINs.

John Daugman has been a Lecturer since 1995 in the Lab, where he teaches courses in Computer Vision, Neural Computing, Information Theory and Coding, and Continuous Mathematics. He obtained his degrees at Harvard University, where he also joined the faculty, and was then awarded the Presidential Young Investigator Award by the US National Science Foundation. Before coming to Cambridge he held the Toshiba Endowed Chair in Computer Science at the Tokyo Institute of Technology. He received the 1997 Information Technology Award of the British Computer Society, and his invention for iris recognition was designated a Millennium Product by the Design Council.

Ian Leslie: *What about the future?*

Rather than getting involved in a crystal ball exercise which will only be of embarrassment to me in years to come, I would like to point out some areas of current excitement in the research that is going on in the Laboratory. I hope I am forgiven for beginning by looking at my own group's activities before moving further afield.

In my own area of research, namely networks and operating systems, there are a number of general themes, including convergence of communications and computing, the increased reliance we all place on systems, and the opening up of network control. They are of course interrelated and I would like to take a brief look at their implications. This is perhaps to take a view of the subject from above.

Looking at things in a bottom up fashion, base level technology – in particular semiconductor technology – improves at a reasonably predictable rate. So do somewhat "higher-level" devices such as processors, memory chips and lasers. But these types of device have different rates of improvement, so system architectures optimised for current device characteristics will reveal fault lines when scaled up to go faster, cheaper, and bigger. Herein, however, also lie opportunities: these range from new workstation architectures to the infrastructural glue that keeps global distributed systems running smoothly and securely while continually evolving.

These developments also create discontinuities – to borrow John Taylor's phrase – of other sorts, in particular with respect to human machine interaction. Stepping back one could simply regard this as another fault line in current architectures; humans aren't advancing in processing and perception power as fast as computers are. In fact, the nature of human machine interaction is under profound change. Exciting as this is, there will be even more exciting developments as we begin to examine the applications which become possible. Some of these applications lie beyond what we may consider computer science; we now see convergence of content with communications and computation.

Interestingly, success in my part of the subject (where success might be gauged in terms of getting ideas into use in the real world) brings more pressure to bear on the foundations of the subject. Society is becoming more and more reliant on technology much of which has come to market very quickly, and this trend will continue. The Y2K bug is only an indication of the problems ahead if we do not have a better understanding not just of how systems usually behave, but of the range of their possible behaviours. Only with this fuller understanding we will be able to develop better means of building more robust systems.

The Laboratory is pursuing research in all of these areas. I hope to be able to give a taster of the immediate future of the Laboratory's research.

Ian Leslie began a PhD in the Laboratory in the area of computer communications in 1978. He became a research assistant in 1981 and joined the teaching staff of the Department in 1983, becoming Professor of Computer Science (Teaching and Research) in 1998. His research interests include operating systems, networks and more particularly their resource management and performance.

Karen Spärck Jones: ***Conclusion – getting there***

Systems have always been the focus of the Laboratory's research. But what 'systems' are, in form and function and in theory and practice, has continually grown. The Laboratory's work is thus protean. The talk will illustrate the rich variety, from self-timed circuits, to proof of security protocols, to spoken document retrieval, of the research being done as the Laboratory heads for its next fifty years in computing.

Karen Spärck Jones was a vicarious EDSAC 2 user (via Roger Needham), then an early Titan user. A research fellow and associate in the Laboratory from 1968-88, working on natural language and information processing. Appointed to staff in 1988, now a Reader. Organiser, EDSAC 99 celebration.

Historic objects exhibition

Overseer: Chris Hadley

The EDSAC 99 historical exhibition shows objects associated with the Laboratory's major machines, focussing on the EDSAC 1. It covers pre-EDSAC calculating machines, EDSAC 1 objects, and then items associated with EDSAC 2, Titan, the CAP, the Cambridge Ring, the Cambridge Model Distributed System and the Rainbow display, leading to the present Autostereo display.

Along with machines, components and devices the exhibition includes documents – the EDSAC log book, design drawings, early project lists, Proceedings of the 1949 Cambridge Conference, etc. – together with photographs, especially of EDSAC 1 and EDSAC 2, with early Laboratory staff. We are particularly grateful to David Wheeler for helping with the identification and description of these artefacts.

These documents are supported by illustrative material showing, for example, the EDSAC 1 order code, early Laboratory PhD thesis titles and Diploma dissertation topics.

The companion exhibition on the Computing Service takes up the theme of service provision and development, begun with the Mathematical Laboratory's original calculating machines and enhanced through the EDSAC 1 projects.

Millionaire calculating machine

An example of a very early calculating machine, from about 1900. The mechanism was unusual for the time in that it had a multiplication unit – it was more common to perform multiplication as a sequence of additions.

Brunsviga and Double Brunsviga hand calculating machine

These are Brunsviga hand calculating machines. They were used in the Laboratory from about 1937 until the early 1950s. After EDSAC started they were used for teaching classes in numerical analysis. Originally the Lab was set up so there were some calculators like this to provide computing for the University, and science students used such machines to do their calculations. A number can be set by means of the keys, which can then be added into the accumulator by turning the handle, or subtracted by turning it the other way. In addition the accumulator at the bottom will move to the right and to the left, which allows a multiplication as a sequence of additions and shifts (or division by doing the same backwards with subtraction). The double Brunsviga was designed to perform calculations with complex numbers.

Facit hand calculating machine

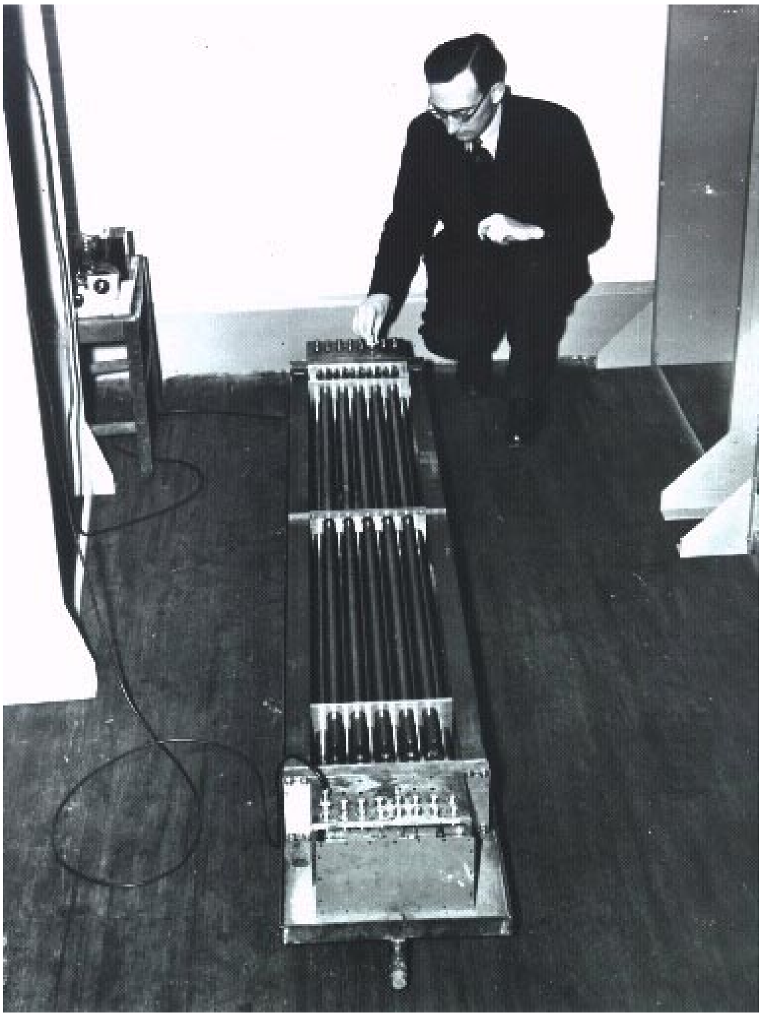
The Facit hand calculating machine dates from the 1950s, and their main use in the Mathematical Laboratory was for teaching numerical analysis. The mode of working is approximately the same as the Brunsviga with some refinements.

Marchant electrical calculator

Marchant electrically driven hand calculators were introduced into the Mathematical Laboratory during the war. They were part of the University Computing Service for use by members of other departments. The mode of use is similar to that of the Brunsviga and the Facit, but the operations are automated rather than driven by hand.

Mathematical Laboratory door

This was the door to Corn Exchange Street from the old Mathematical Laboratory. It was unofficially saved and has become a historical relic recording retiring members of the Laboratory. It was mentioned in Fred Hoyle's science fiction book "The Black Cloud".



MV Wilkes & EDSAC 1 mercury delay lines

EDSAC 1 log book

The EDSAC 1 Log book was started on 6 May 1949. The machine had been working and been commissioned before then but no records were kept and so authentic dates cannot be given. It started with an informal system and proper computing didn't start until about three months later. The first program printed a table of squares and was completely automatic. In other words the program was read in from punched paper tape, it did a computation of two minutes thirty seconds, including printing the results on a teleprinter – the whole was automatic apart from pressing the start button.

EDSAC 1 chassis

This is a chassis from EDSAC 1 which had fifteen racks, each of which held five to ten chassis. These were wired in permanently. The wiring was underneath and the back wiring connected chassis across the full room which was about 15 ft by 15 ft. The purpose of this particular chassis is unknown but a chassis of this size would typically run one memory tank in the memory storing 16 words, or one word of an accumulator, or half a word of the instruction register. Racks of similar size were used to make adders and other selection devices. In the very early days on the memory chassis the memory was cleared by touching a diode at one end – although later a press button was fitted to clear the entire store. It was manufactured by Morley and Duke, an electrical wiring workshop here in Cambridge, after the prototypes had been made in the Mathematical Laboratory.

Creed paper tape reader, used with EDSAC 1

This item is a paper tape reader quite standard at the end of the war – it was used in telegraphy. It was adapted for use in the tape room although as it only works at 6 characters a second it was later replaced by tape readers manufactured in the Mathematical Laboratory. It was used for preparing paper tapes to be fed into the EDSAC, but was not actually physically connected. In preparing a program a user might wish to incorporate a library routine. It was possible to copy a library tape and part of the program tape and assemble a complete program tape from these individual parts – and so the tape room contained a number of tape readers and punchers. The output was a standard telegraph serial code – EDSAC 1 had a parallel input of 5 bits so it could not be used there but it could be used as a tape reader to copy and drive a similar punch.

EDSAC 1 mercury delay line

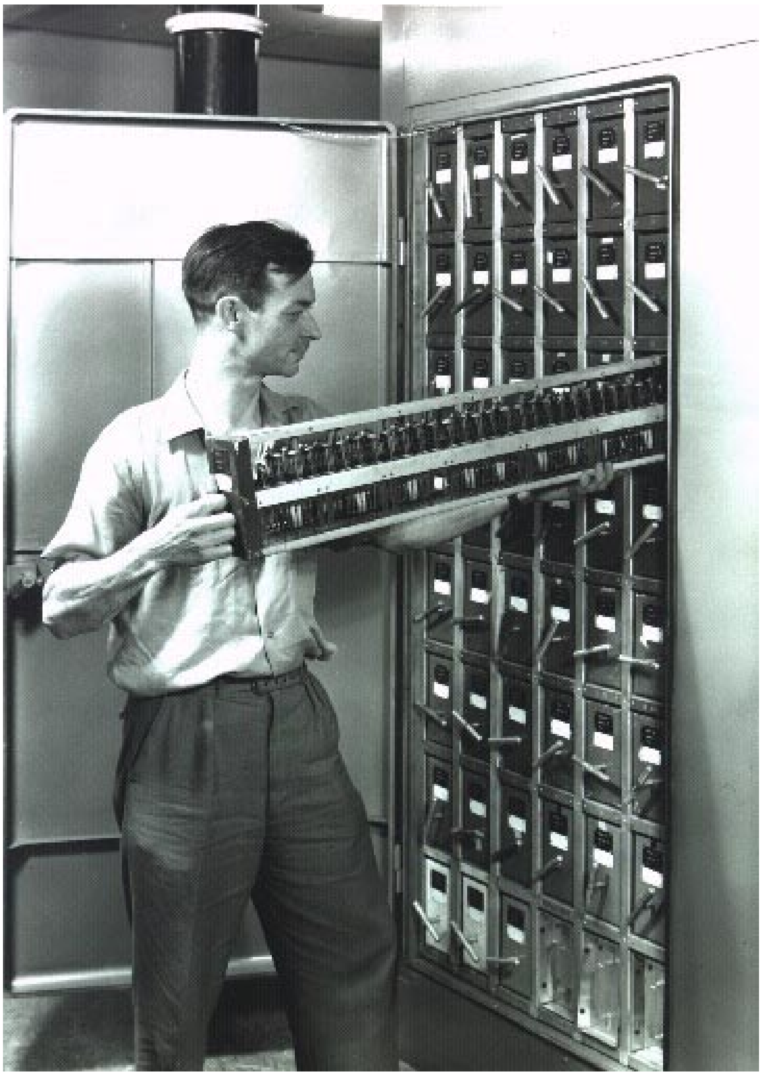
This is a mercury delay line, used as memory in EDSAC 1. The main memory consisted of two batteries of 16 tanks, each 6 feet long. This particular delay line was used for holding one word of 35 bits. An oscillator drove a quartz crystal transducer at one end. The sound waves travelled through the mercury within the tube and were picked up by a quartz transducer at the other end. The length is such that 35 pulses could be travelling through the tube at any one time, each about one microsecond long, sent every two microseconds. There was a circuit in a chassis which took the signal coming out, amplified it, and converted it into a voltage pulse. This was then ANDed with a clock pulse and sent back, or possibly, according to control wave forms, some new information was fed in – so it could hold the information as long as was necessary. The attenuation was so small on the ultrasonic pulses that it was possible for a pulse to be reflected from the far end which would cause interference – there is a screw at the top which could be screwed into the mercury to increase the attenuation which prevented this happening. The six foot long tanks did not need this increased attenuation. The tube was filled with mercury because this happened to match the characteristics of quartz crystals transducers, so there was a good impedance match.

EDSAC 1 thermostat

This is a thermostat, made from a mercury thermometer. At the appropriate height of mercury two circuits shorted and this was used to control the temperature of the mercury delay lines used by EDSAC 1. The delay lines were put in a coffin-like box, and the thermostat was arranged to keep the temperature constant so the number of pulses in each delay line remained invariant. It was made in the Lab by hand, there being no commercially available devices available at the time.

EDSAC 1 valve lifter

A device, made in the Mathematical Laboratory, for lifting hot valves during operation.



SA Barton & EDSAC 2 chassis construction with miniature valves

EDSAC 2 chassis (various)

EDSAC 2 was made of a number of such plug-in units, the modular construction allowing easy maintenance. The bulk of the machine consisted of 41 identical long chassis, each of which was one part of the arithmetic unit and contained six one bit registers, an adder, a complementer, and connections to the rest of the machine. There were other shorter chassis each of which controlled one bit of the magnetic core memory. The form of construction is such that air could be blown through sideways to keep the components cool (in contrast to the EDSAC 1 which relied on natural circulation of air round the valves). The screw at one end and turn knob at the other were used to physically force a plug into a socket to make reliable contacts.

Paper tape reader, used with EDSAC 2

This is a photo-electric tape reader which was developed at the Mathematical Laboratory. It could read in paper tape at a thousand characters a second with the ability to stop on a single character. It has a motor which revolves a cylinder which is pressed against the tape to pull it forward – there is a brake which is actuated by electromagnet underneath and which is strong enough to stop the tape in spite of the rotor pulling it forwards. There is an essential button on the top which allows the tape to be ejected at full speed without being read. This is only capable of reading a 5-hole tape unlike others which were developed from this by Elliots and the Engineering workshop. These were first used with EDSAC 2 but continued in use until the 1970s.

EDSAC 2 engineer's control panel

This control panel allowed an engineer to run the computer, to do single orders, and to do single micro-steps. It was at the back of the machine and indicator lights enabled the engineer to trace the course of the register transfers or computation. There was a speed control, and volume controlled loudspeaker.

Magnetic tape holder, used with EDSAC 2

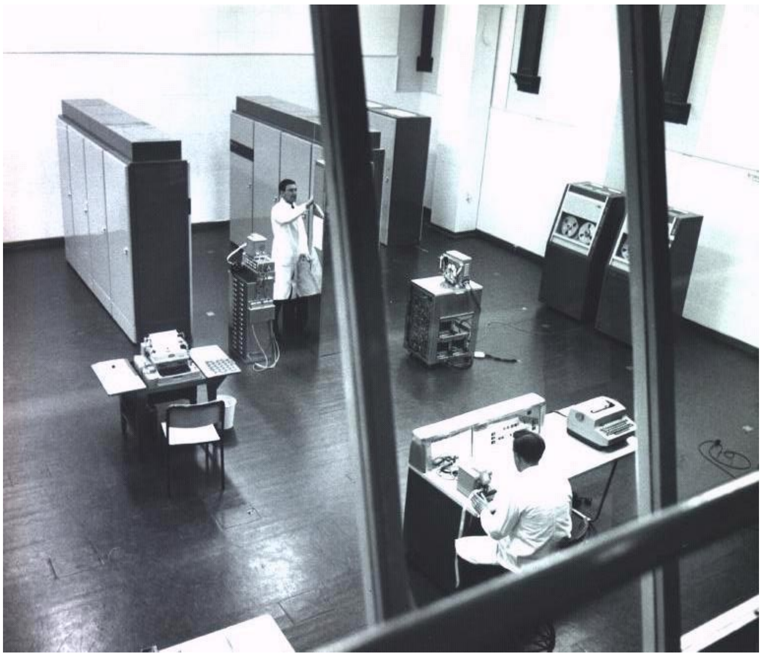
EDSAC 2 used one inch magnetic tape and this was held on reels. One of the interesting aspects of this magnetic tape system was that the heads ran out of contact with the tape, resulting in theory in indefinite life but less resolution. The tapes were stored in these metal boxes to prevent magnetic influence affecting the tapes. This particular reel originally had the autocode compiler for EDSAC 2. The DECCA device was partly developed in the Lab but manufactured by DECCA.

Creed tape perforator, used with EDSAC 2

This is a keyboard perforator made by Creed. In normal use the keyboard is used and it produces a paper tape from a roll with up to 5 holes punched across the tape. Extra solenoids were added which could actuate the punches, and they were used as a fast output for both EDSAC 1 and EDSAC 2, running at about 35 characters per second. This particular one was connected to EDSAC 2. They were also used in the tape preparation room for copying tapes. Eventually they were replaced by a high speed Creed punch and teletype (110 characters a second).

EDSAC 2 control matrix component and production moulds

The control matrix housed the ferrite cores which stored the microprogram for EDSAC 2. This plastic housing contained 16 13mm cores and associated wiring, 64 such units were needed (in an 8 x 8 flat array), making 1024 cores in all. Smaller core memories were used in other parts of EDSAC 2.



Titan under construction (from the public gallery)

Titan console

This console was the main operating console of the Titan. In practice it was only used by the engineers.

Titan card

One board from the Titan, which was essentially a slightly cut down Atlas computer. The Atlas was made of boxes each of which had fifty such boards in, and altogether there were 10 boxes to a rack and 3 racks in the case of the Titan. It illustrates the circuit techniques of the time – using standard transistors, individual diodes, individual resistors – i.e. transistors had been invented but integrated circuits had not. The machines could be repaired easily due to interchangeability of parts. The different board types were numbered according to the well-known resistor colour code (the three plastic tabs on one side).

Magnetic disc for Titan

This is a magnetic disk supplied by Data Products for the Titan computer. The entire unit was made of 16 or 32 of these. They utilised moving magnetic heads similarly to today's disk drives. This was the bulk memory for the Titan and it stored quite a large amount of data for the time. It stored data in 512 word blocks like a magnetic tape. Products like this were subject to destructive contamination – if the magnetic head touched the surface it scraped off magnetic particles which then came between the read/write head and the disk and caused more particles to be created, thus leading to the catastrophic destruction of the disk. This happened occasionally. Data Products were an American company who specialised in printers and disk stores.

Disc controller board from Titan

This is one of the boards in the Data Products disk controller used with the Titan computer. It contained a large number of magnetic reed switches which were used to position the arm and these were controlled by transistors and coils. This was just one of the components of the Titan disk unit, made by Data Products in California.

Ferrite core memory

This is a typical ferrite core memory plane. It contains 32 x 32 cores with 3 wires through each core. These were the first reliable fast memories made for computers. Every single bit was stored on a ring of ferrite and there were three wires passing through with an X drive, a Y drive and a sensor, and if both the X and the Y drive were applied the core would switch – the third wire picked up the signal. The cores had an inherent square loop property which made this possible, if you only half disturbed the core it did not switch magnetisation but if you gave a full write with both X and Y it would switch magnetisation. The mode of operation was to read a bit and if a signal came out that bit would be written back – it was a so-called disruptive read. It took about two decades before they were superseded by integrated circuits because the ferrite and manufacturing process became steadily cheaper. In the 50s they were hand wired and then later they were wired in the Far East – their size reduced and their speed increased and so they lasted a surprisingly long time.

Titan B ferrite core store

This unit was used on the Titan – a so-called B-core store. It has 128 words of 24 bits and was used for storing the B-registers associated with the order code. It had for the time a very rapid access of about half a microsecond, and as it stored a fair number of registers it allowed versatility to the program. It was fairly intensively used so the coils would tend to get a little hot. They were working at the highest speed possible and oil was put in to allow for heat dissipation.

Elliot tape reader

This is a tape reader made by Elliot which was copied from one designed in the Computer Lab. The principles are very simple but it went considerably faster than other tape readers of the time. There was a continuously running roller and underneath that a pinch roller operated by electromagnet and when these two closed the tape was drawn through the tape reader. A slab of metal resting on top of the tape could be attracted by an electromagnet underneath and when this happened it applied a frictional force and the tape stopped immediately. It could run at a 1000 characters a second and stop within half a character distance (a twentieth of an inch). An interesting part of the design is the brakes. When it was originally tried it was discovered that under certain conditions it set off enormous vibrations in the tape reader and so didn't work. It was also discovered that by laying a finger on the top it would then work. The finger is replaced in this design by a little slab of foam between the two parts of the brake which serves the same purpose. Another part of the design was a push button which caused the rest of the tape to be ejected at high speed from the tape reader. Part of the design isn't here – some means of collecting the tape which is spinning out at high speed is needed, for which large empty cardboard boxes proved suitable.

Rollerball tracking device

This device which was made for moving a point on a screen in a similar way to a trackerball. The yellow sphere would move the pointer on the screen, and it can be thought of as an upside-down mouse. These roller balls still exist on some computers today but this one was designed and built in the Lab in the early 60's. It is one of several novel devices developed here to be attached to graphical display devices.

PDP-7 joystick

This was used for controlling the graphical pointer on a graphics display which was driven by the PDP-7 connected to the Titan. It was mainly used on its own and could be used for graphics using a relatively large screen. The joystick is roughly equivalent to a mouse in function, being used to control a pointer on the screen (although a mouse wouldn't usually have a rotational movement). It was made in the Lab, and was used throughout the life of the PDP-7 (approx 1966-73).

Experimental graphical input for PDP-7

This experimental graphical input for the PDP-7 was an attempt to move a pointer around a Cathode Ray screen in a similar way to a mouse. It used a simple ferrite pickup with cross wires to detect the location of the pointer. It was not in use for very long as other such devices appeared commercially.

Titan multiplexor

This allowed 64 lines from teletypes to be accessed by Titan. When a character was received an interrupt was given to Titan which could then read the character and the source number. The Titan could also send characters to each destination. The multiplexor was essential for online access across the University.

Titan plaque

This was produced as a demonstration of the foam cutter connected to Titan which could carve out 3 dimensional objects under computer control. The foam was sometimes used to form a cast in which metal could be poured to make metal objects.

RM Needham & DJ Wheeler explaining the Ring to the Chancellor

The CAP

The CAP (Capability Protection) Project ran from 1970 to 1977. It was an experiment in memory protection, based on capabilities implemented in hardware, under M.V. Wilkes and R.M. Needham with D.J. Wheeler responsible for implementation (BCS Technical Award 1978 for 'CAP (Capability Protection) Project' to R.M. Needham).

Control panel from the Philips store

The Philips core store was a later add-on to the CAP. It provided 120k 32-bit words, but was rather slow as the words were byte addressed.

Paper tape library

A selection of paper tapes containing subroutines which could be physically copied and stuck together to form parts of larger programs. This technique was common to most computers until the 1970s.

Cambridge Ring repeater

This is one component of the Cambridge Ring, a repeater. The Cambridge ring consisted of a pair of twisted wires which were taken round the Lab and they plugged into one of these devices – this allowed some device which accepted or gave information to be attached to the Ring. This device would regenerate the pulses. A particular Ring would have up to 256 such repeater elements. These were designed in the Lab but they were wire-wrapped by an outside firm. The Cambridge Ring was supplied to many universities throughout the UK – the last one to be used was at the University of Hertfordshire, disconnected only 2 or 3 years ago. The project started in 1974.

Cambridge Ring monitor station

This object is the ring monitor station, an essential part of the Cambridge Ring. It supplied power to all the repeaters around the Ring. It provided start up facilities and monitored the Ring for accuracy and errors. Typically it was switched on and left working and it was possible to perform some tests on the Ring from this panel. Access to the printed circuit board containing the components was behind the sliding panel.

Ring circuits (experimental)

Experimental boards produced during the development of the Ring.

Ring PCB

A circuit board connected with the Cambridge Ring. Later versions of the Ring had more integrated circuits – in fact one circuit was designed by Andy Hopper to replace nearly all the chips on the previous boards. This was made to work but it never came into service here because it would only run safely at about 7 MHz and not the 10 MHz specified, mainly due to doubtful specifications by manufacturers. The large transformer is used for generating the power supply for the board from the Ring itself. Repeaters took power from the Ring so that they would be independent of the units connected to the Ring, other stations took power from their host device.

Ring data logger

This was part of the original experimental Ring. This was a data logger – it would detect errors, count them and display them, and hence was used for diagnostics and to assist in maintenance. Each station would automatically signal back to the monitor station if it detected a parity fault, and the monitor station would cause it to be logged. In practice the Ring did not need much maintenance.

Ring keyring

A chip implementation of the Cambridge Ring was developed as part of a formal project under the Government's Advanced Computer Technology Projects. Some of the resulting chips were later made into commemorative key-"rings".

Type 1 microprocessor board

Small servers in the Cambridge Distributed Computing System were based on the Z80 microprocessor. This is this first production version of the "Type 1", with 4K of static RAM and 1K of EPROM. The application code, typically programd in assembly language, was downloaded over the ring interface. These systems were often used as peripheral drivers, and an area of the wire-wrapped board was left blank for special purpose interface hardware to be added.

VDU concentrator board

The Z80 processor was also used to make terminal concentrators to connect simple character cell VDUs to the ring. A user of a ring terminal concentrator could connect to any machine on the ring and run several simultaneous sessions. The board shown here has UARTs and line drivers to connect eight terminals. This board was connected to a Type 1 processor by a short ribbon cable. All of these boards were later redesigned on printed circuit boards.

The Rainbow display

The Rainbow display was an experimental terminal to support graphical interaction with the Cambridge Distributed System. It is built around one of the Motorola 68000 computers that were also used in the main processor bank. This has two peripheral controllers based on 6809 processors, one a standard "Type 2" system controlling the ring interface and the other handling the keyboard, mouse and tablet. The second crate contains the display hardware itself. This consists of a bit-slice micro-controller built around AMD 2901 chips, 64k 16-bit words of video memory and an output pipeline that assembled a video signal in real time for windows stored in different parts of the memory. (BCS Technical Award 1985 to N.E. Wiseman.)

CK Hadley & mercury delay line

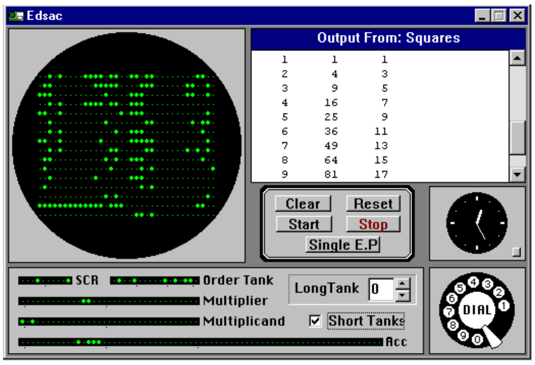
"Some people could not believe that the mercury memory would ever work."
[M.V. Wilkes, Memoirs of a Computer Pioneer, MIT Press, 1985, p128.]

EDSAC simulators

Overseer: Simon Moore

Students in the second year of the Computer Science Tripos undertake group projects. This year two teams wrote EDSAC simulators in Java, following the earlier example of Martin Campbell-Kelly at the University of Warwick. All three are shown here.

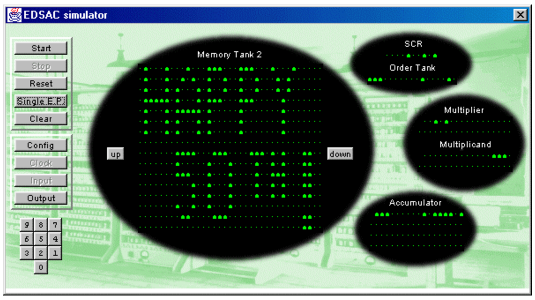
Martin Campbell Kelly's original EDSAC simulator is a faithful software evocation of the EDSAC computer as it existed in 1949-51. The user interface has all the controls and displays of the original machine, and the system includes a library of original programs, subroutines, and debugging software. The simulator is intended for use in teaching the history of computing; as a tutorial introduction to the classic "von Neumann" computer; or as an historical experience for current computer practitioners.



```
  1     1     1
  2     4     3
  3     9     5
  4    16     7
  5    25     9
  6    36    11
  7    49    13
  8    64    15
  9    81    17
```

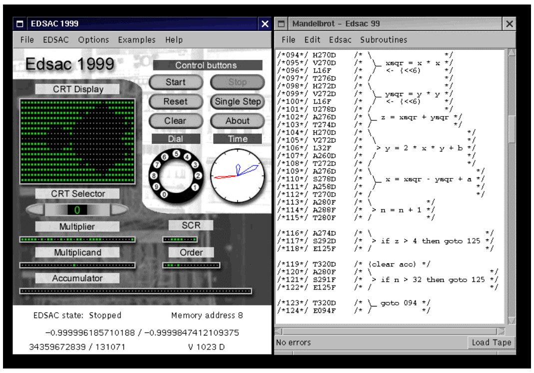
//www.dcs.warwick.ac.uk/~edsac/

Group Echo consisted of: Matthew Bentham (documentation), Trevor Boyd (research, testing & example programs), David Dunwoody (management, documentation & Web site), Crispin Flowerday (CPU simulator), Simon Frankau (simulation kernel), Simon Freytag (graphical interface) and Simon Greenway (interface code).



//thor.cam.ac.uk/group/CST1b/echo/

Group India consisted of: Joe Marshall (CPU simulator), Alan Mitchell (project management & documentation), Dominic Penfold (interface design & Web site), James Slorach (testing & example programs), Chris Town (testing & example programs), Matthew Wakeling (code librarian) and Colin Watson (graphical interface).



```
EDSAC state:  Stopped              Memory address 8
   −0.999996185710188 / −0.9999847412109375
34359672839 / 131071                   V 1023 D
```

```
/*094*/ H270D    /* \              */
/*095*/ V270D    /*  \_ xsqr = x * x */
/*096*/ L16F     /*  /   <- (<<6)    */
/*097*/ T276D    /* /              */
/*098*/ H272D    /* \              */
/*099*/ V272D    /*  \_ ysqr = y * y */
/*100*/ L16F     /*  /   <- (<<6)    */
/*101*/ U278D    /* /              */
/*102*/ A276D    /* \_ z = xsqr + ysqr */
/*103*/ T274D    /* /                */
/*104*/ H270D    /* \                     */
/*105*/ V272D    /*  \                    */
/*106*/ L32F     /*   > y = 2 * x * y + b */
/*107*/ A260D    /*  /                    */
/*108*/ T272D    /* /                     */
/*109*/ A276D    /* \                       */
/*110*/ S278D    /*  \_ x = xsqr - ysqr + a */
/*111*/ A258D    /*  /                      */
/*112*/ T270D    /* /                       */
/*113*/ A280F    /* \          */
/*114*/ A288F    /*  > n = n + 1 */
/*115*/ T280F    /* /          */

/*116*/ A274D    /* \                       */
/*117*/ S292D    /*  > if z > 4 then goto 125 */
/*118*/ E125F    /* /                       */

/*119*/ T320D    /* (clear acc) */
/*120*/ A280F    /* \                        */
/*121*/ S291F    /*  > if n > 32 then goto 125 */
/*122*/ E125F    /* /                        */

/*123*/ T320D    /* \_ goto 094 */
/*124*/ E094F    /* /          */

No errors
```

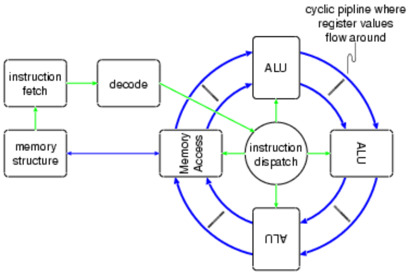
//thor.cam.ac.uk/group/CST1b/india/

Current research

Overseer: Graham Titmus

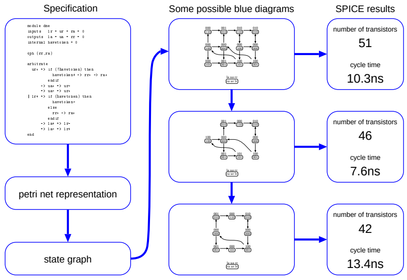
Self-timed logic (Simon Moore & Peter Robinson)

The increasing complexity of integrated circuits raises difficulties in ensuring logical correctness, high speed and good manufacturing yield. Particular problems arise with timing which inhibits the composition of proven modules into a larger whole. *Self-timed logic* offers a solution but requires new approaches to design at all levels.

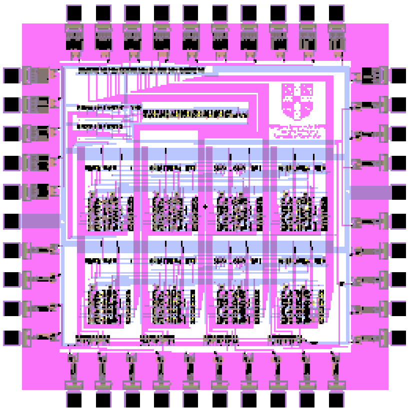


The *rotary pipeline processor* is a new architecture for superscalar computing. It is based on a simple and regular pipeline structure which can support several ALUs for efficient dispatching of multiple instructions. Register values flow around a rotary pipeline, constrained by local data dependencies. During normal operation the control circuits are not on the critical path and performance is only limited by data rates. It is particularly suited to self-timed implementation.

Self-timed control (handshake) circuits have highly concurrent behaviours which make them complex to design. We have developed a new diagrammatic technique – Blue Diagrams – for analysing self-timed control structures. This is being used as the basis of a handshake circuit synthesis tool.



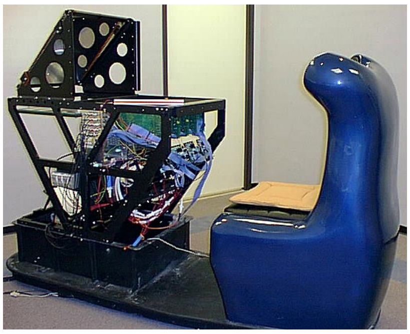
Transistor level design issues for rotary pipelines have been undertaken. This led to the development of a new flip-flip – a RED (reset, enable, data) flip-flop – for use with dual-rail encoded dynamic CMOS logic. This flip-flop also required new control structures to take advantage of its unique characteristics.



A full custom CMOS test chip has been manufactured to test a number of different pipeline structures based on RED flip-flops and new control structures.

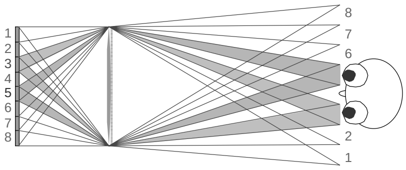
We (Moore, Robinson, Taylor & Wilcox) are now working with support from two local companies on applying these techniques to micro-controller design, with a particular emphasis on low powered operation. This involves further research on simulation, synthesis and compiler technology.

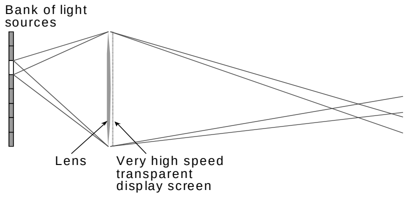
Autostereo 3D display (Neil Dodgson & Stewart Lang)

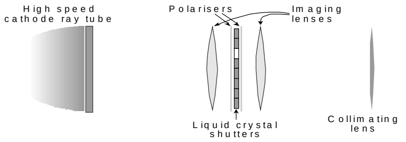


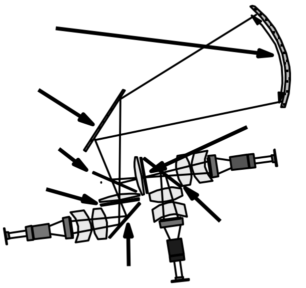
The autostereo display gives a truly 3D picture without the viewer needing to wear any special glasses. It was developed jointly by the Computer Laboratory and the Department of Engineering. It is in the process of being commercialised by ASD Systems, one of the many companies spawned by the Laboratory.

Human beings see the world through two eyes. Because the eyes are 65mm apart, each sees a slightly different image. The brain combines these two images, in a process called stereopsis, to give us depth (3D) information. With a conventional television, both eyes see the same picture, so you do not get stereopsis. A 3D display provides a different image to each eye, and the human brain does the rest.









Time-multiplexed autostereo

Most 3D displays provide a different image to each eye by forcing you to wear special glasses. The Cambridge display produces the 3D effect in a different way. It displays several different images, each taken from a slightly different point of view. The images are displayed very rapidly, one after another, and internal optics ensure that each image is visible in only one of a set of viewing zones in front of the display. Each eye will be in a different viewing zone and hence will see a different image, providing stereopsis without special glasses.

Commercialisation

Displays like this have uses in visualisation of complex 3D structures, in remote manipulation, and in entertainment. ASD Systems have built two commercial prototypes: a 25”, 28 view display and a 50”, 15 view display. These are currently being tested by a variety of international companies.

Continuing research (Druti Shah & Andy Penrose)

ASD Systems continue work on improving the display hardware. This technology requires an order of magnitude more image data than a conventional display and so, in the Computer Laboratory, we are concentrating on the problems of image compression and image generation.

Next generation workstations (Ian Pratt & Austin Donelly)

The majority of today's computers are designed around a central shared bus interconnecting the computer's CPU and its various peripherals such as the network, disks and display. In contrast, the Desk Area Network (DAN) workstation interconnects devices using a space-division switch. The switch enables data to be transferred between devices at substantially higher bandwidth than is achievable with traditional bus-based designs. Furthermore, multiple 'conversations' between different devices can be in progress simultaneously, enabling even greater aggregate bandwidth to be achieved.

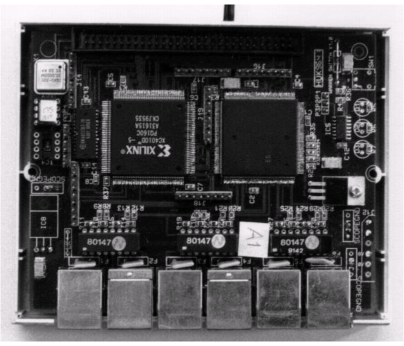
In conventional systems, applications wishing to perform I/O must ask the operating system to perform the operation on their behalf. This is necessary since the operating system takes responsibility for multiplexing the device between different applications, and ensuring that they do not violate system integrity or attempt to access data to which the user is not entitled.

In comparison, devices within the DAN Workstation are specially designed to enable them to take on some of this responsibility themselves. Thus, applications are able to talk *directly* to the device when performing most common types of I/O operation (e.g. sending and receiving packets over the network, reading and writing disk files, drawing on the display).

Bypassing the operating system means we can gain increased throughput and reduced I/O latency. This is particularly useful when performing distributed computation across multiple workstations, or when processing continuous media streams such as digital audio and video.

Furthermore, DAN devices can be intelligent enough to setup and manage data transfers between themselves, off-loading work from the main CPU and leaving it free to concentrate on computation rather than I/O. Features like those found in the DAN Workstation look set to become commonplace in future commercial systems.

Warren home area network (David Greaves & Daniel Gordon)



The Warren is a network for the home which connects household devices together to allow video and audio to be transferred between them, as well as allowing these devices to be controlled. IR remote-control receivers in various devices can be pooled and information sent to a central CPU, allowing every device to be controlled from anywhere in the home using an IR handset. Internet traffic can also be routed around the home. The use of ATM allows for video, audio, control, IR and Internet traffic to be integrated together and sent around the home on just one set of wires.

The Warren is an ATM subnetwork composed of an arbitrary mesh of very simple ATM switches and end stations and which is connected to the outside ATM network at one or more points. The novel aspect is that switches and end stations within the Warren are implementable entirely in hardware – i.e. they require no microprocessors for signalling and management. All of the signalling software for all the devices within the Warren is provided by one or more proxy servers. The proxy servers are software entities situated on one or more general-purpose computers connected to the external ATM network outside the Warren.

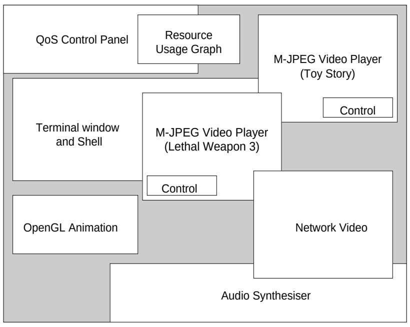
Warren devices

Some of the Warren devices are existing household appliances which have been altered by adding an ATM interface, and others have been built from scratch. Current devices include an ATM Switch, Phone, Microphone, ATM to Stereo Module, Compact Disc Player, Infrared Base Station, LCD Display Tile, Warren Controller and many others.

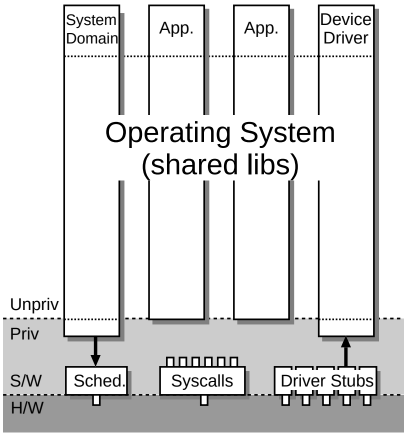
This project is being carried out at the University of Cambridge Computer Laboratory with the support of Virata Ltd, Cambridge. Further information can be found at http://www.cl.cam.ac.uk/Research/SRG/HAN/Warren/.

The Nemesis operating system (Steven Hand & Paul Menage)

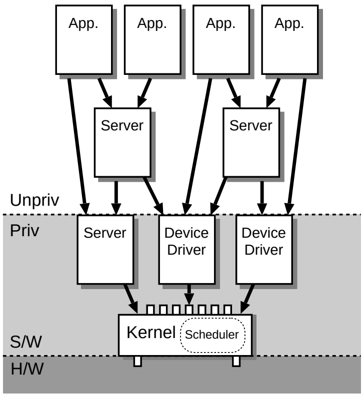
The Nemesis Operating System has been developed under the Pegasus and Pegasus II ESPRIT projects to provide applications with fine-grained guarantees for all of the resources which they consume. In traditional operating systems, tasks such as network protocol processing, paging, and graphics rendering are performed either in a shared server or in the kernel – unless this work is carefully scheduled and accounted for, it can cause Quality of Service interference between applications.



Nemesis aims to simplify such accounting by enabling each client to perform much more of their own work than in a traditional system; shared servers (or the kernel) are needed only to arbitrate access to shared resources. Wherever possible, these servers exist purely for setting up and destroying access connections (e.g. providing access to the blocks on a disk constituting a file, or installing a packet filter to demultiplex network packets). Other operations such as network protocol processing or graphical rendering are performed in the application itself (typically using shared libraries), with the servers just providing safe scheduled access to the hardware.



Nemesis Structure

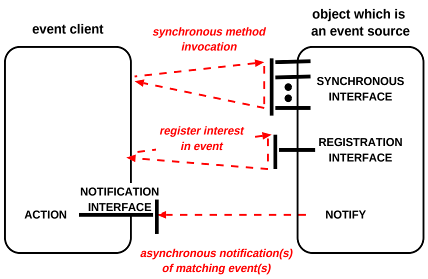


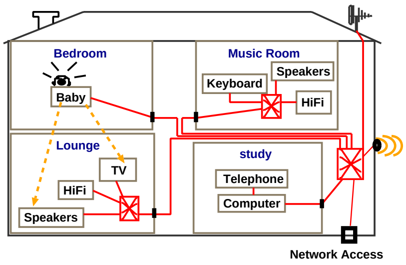
Traditional Microkernel Structure

This philosophy allows Nemesis to provide an effective platform for “multi-service” systems, where different multi-media applications co-exist without experiencing interference from one another or from other applications with less critical time constraints.

The active house (Jean Bacon, Andrew McNeil & Alexis Hombrecher)

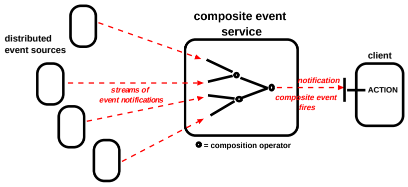
The Cambridge event architecture is based on a publish, register, notify paradigm. An event is a parameterised, asynchronous message, which is an instance of an event class. Objects, which are event sources, publish the event types they produce. Other objects register interest in these events, possibly providing parameter values to be matched. When an event occurs, all registered parties with matching parameters are notified and may take appropriate action.


The *publish-register-notify* paradigm

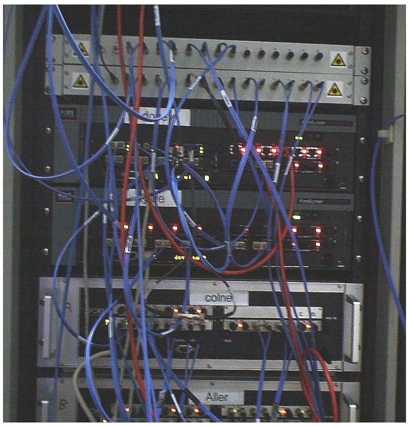

An Active Home

The *active house* demonstration uses events to link a range of automated appliances within a virtual house. The appliances can both produce and receive events; i.e. they can act as an event source and sink at the same time, and publish, register, notify and receive events as described above. On receiving an event an appliance will typically perform some action.

The owner of the house may wish to express quite complex policies on how it should behave in terms of combinations of event occurrences. We provide a simple composite event algebra for this purpose, allowing more complex and more useful interaction between event sources and sinks.


Event Composition - Composite event detection

Network control & management (Ian Leslie & Richard Mortier)

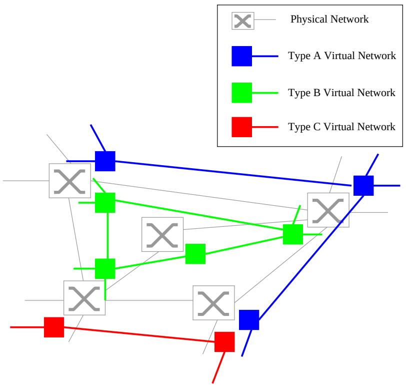


Asynchronous Transfer Mode (ATM) is an advanced networking technology allowing diverse traffic streams to share a single physical link. It provides a Virtual Circuit service, allowing data entering the network to be switched at high-speed to its destination. Setting up virtual circuits is referred to as the Control Plane of the network, and the software that carries this out a Control Architecture (CA).

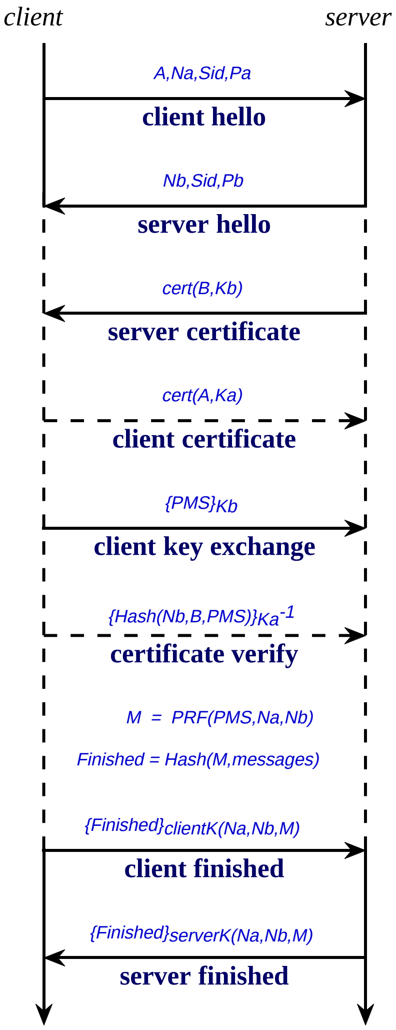
ATM control architectures have developed from those used in the telephone network. They tend to be monolithic, highly complex, and inflexible. This makes it hard to upgrade them when new network services are designed. Additionally, most systems run only a single CA, so there is no provision for supporting multiple CAs at once. Allowing multiple CAs would enable service-specific optimizations to take place and could lead to a more efficient use of the network.

However, there is a solution to these problems! It falls into two parts; first, providing a generic switch control interface to CAs, to allow them control of the resources on the switch; second, partitioning the resources of the switch to allow multiple CAs to operate concurrently.

This partitioning is “hard”, meaning that CAs are incapable of interfering with each other and that it is possible to provide “Virtual Private Networks” (VPNs) sharing the same physical network. Each of these VPNs has a partition of the total resources of the physical network allocated to it and may run its own CA. Since the partition is hard, the CA running on a given VPN may do as it wishes with its resources – it will be prevented from using resources in such a way as to affect other VPNs and CAs.



Proving protocols correct (Larry Paulson & Gianpaolo Bella)



Networks are vulnerable. Suppose that Monica sends an e-mail to her friend Bill. Since the message must pass through many computers, other people can read or alter it, and send bogus messages that appear to come from her.

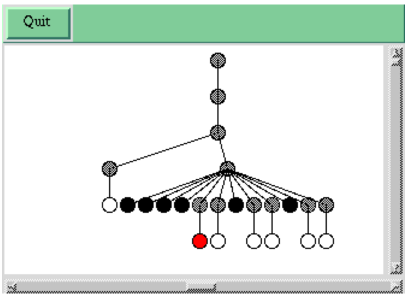
Cryptography can help. Monica and Bill can run a security protocol: a pre-arranged message handshake. (The figure on the left shows one used in Web browsers.) Here Monica and Bill use public-key cryptography, to generate temporary encryption keys. Eventually they have their electronic conversation. Nobody can listen in, and Monica or Bill will notice if the data has been modified.

The protocol is complicated in order to thwart attacks. An adversary could play back old messages or combine parts of different messages. To detect this, Monica and Bill label their messages with random numbers every time they run the protocol. The protocol also tries to minimize the use of slow public-key operations.

Unfortunately, many such protocols contain errors. Below, Monica runs a protocol with Linda, mistaking her for a friend. Now Linda can repeat Monica's messages (with slight changes) to make Bill think he is in contact with Monica. Typically, such attacks do not involve any codebreaking.

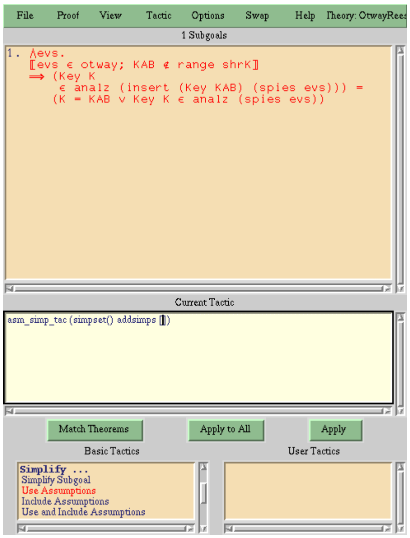
Researchers in the Computer Laboratory (Paulson, Bella, Eastaughffe) have developed new methods of proving protocols correct. They use a simple model of what can occur when honest parties share a network with an intruder. Several industrial-grade protocols have been analyzed, the protocol on the left took about six man-weeks. No other methods can analyze cryptographic protocols in such detail. Can the loss of secrets compromise future sessions? Is timing handled correctly? The proof tool used, Isabelle, was also developed largely in the Computer Laboratory.

XIsabelle – supporting proof (Katherine Eastaughffe)



Automated proof assistants are tools which check the soundness of a person’s deductive reasoning. The user supplies the steps of a proof to the assistant which ensures that only valid chains of inference are developed. They are popular for proving the correctness of programs, hardware designs, high level system designs, security protocols, language compilers as well as certain domains of mathematics.

One such proof assistant is Isabelle, developed at Cambridge by Larry Paulson. XIsabelle is a package that provides extra support to people proving theorems using Isabelle. Providing good support for proving theorems is very challenging from both a technical and a human factors point of view. XIsabelle helps with three aspects of proof support.



```
1. ⋀evs.
   ⟦evs ∈ otway; KAB ∉ range shrK⟧
   ⟹ (Key K
      ∈ analz (insert (Key KAB) (spies evs))) =
     (K = KAB ∨ Key K ∈ analz (spies evs))
```

```
asm_simp_tac (simpset() addsimps [])
```

Multiple Views

For a complex and demanding activity such as theorem proving it is useful to develop more than one model. Different views make different information and manipulation of information explicit and easy to understand.

Search Facilities

There is a huge mass of information from which the user must choose the right pieces to make progress. XIsabelle provides a matcher which finds theorems from a database which might be relevant to the problem at hand.

Undoing, redoing and re-using proofs

Theorem proving typically involves frequently retracing one’s steps, as well as performing similar sets of steps at different points. It is vital that a proof assistant provide good support for these kinds of operations.

Floating point verification (John Harrison & Myra van Inwegen)

It is essential for floating point hardware to get the right answer. Computer owners will not tolerate a processor that gives the wrong answer for ‘simple’ numeric computations.

We are writing programs to check the implementation of floating point hardware. The first algorithm we verified was Tang’s algorithm for computing the exponential function.

Why is this hard?

In order to get reasonable speed and accuracy, floating point algorithms use lots of tricks, including table lookup and polynomial approximations. The resulting algorithm often looks simple, but proving that the result is close to the mathematical ideal is tricky, involving intricate numerical analysis and deep theorems about mathematics.

What was verified?

The main part of the algorithm after special cases have been handled is shown on the right. X is the input and E the output. Several prestored constants and tables are used.

```
if abs(X) < THRESHOLD_2
then
   E := Plus_one + X
else
   N := INTRND(X * Inv_L);
   N2 := N % Int_32;
   N1 := N - N2;
   if abs(N) >= Int_2e9
   then R1 :=
      (X - Tofloat(N1) * L1)
      - Tofloat(N2) * L1
   else R1 :=
      X - Tofloat(N) * L1;
   R2 := Tofloat(--N) * L2;
   M := N1 / Int_32;
   J := N2;
   R := R1 + R2;
   Q := R * R * (A1 + R * A2);
   P := R1 + (R2 + Q);
   S := S_Lead(J) + S_Trail(J);
   E1 := S_Lead(J) +
      (S_Trail(J) + S * P);
   E := Scalb(E1,M)
```

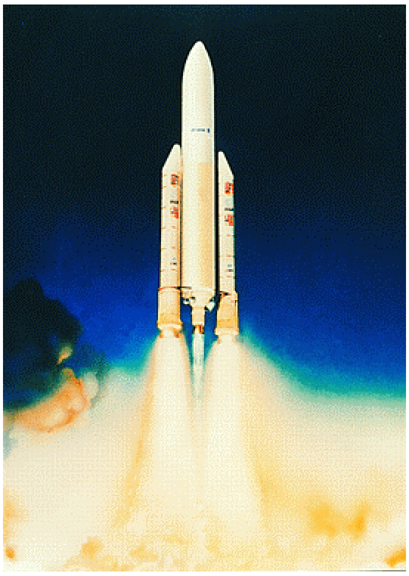
This is written in a simple programming language rather than in mathematical notation. This ensures that we verify something close to the actual implementation. We also wrote a semantics for the language giving a precise meaning to each statement.

How did we do the proof?

We wrote in the HOL logic a statement of exactly what we wanted to prove. Using our Hoare style rules for the language, we figured out what we had to prove for each step in the algorithm. We wrote tactics which automatically proved some of the steps, while extensive work in the field of numerical analysis and real mathematics was required to prove other steps.

The Pentium bug that lost Intel half a billion dollars was a mistake in the floating-point division algorithm. That algorithm, like the exponential algorithm we have studied, uses table lookup, and several of the table entries were incorrect. The methods used here would have found the problem, as the desired error bounds would not have been provable.

Modelling interactive systems (Philippa Gardner)



Interactive systems are pervasive: a cashpoint interacts with a cashcard and the bank, a program accesses the memory, a pilot interacts with air-traffic control and Java code is sent across the Internet. We don't understand interactive behaviour very well. For example, a software problem caused the Ariane 5 explosion. Mathematical models of interaction can help prevent such problems. These models are necessary for rigorous language design, program development and specification analysis.

A simple example of interaction involves computable functions, described formally by Turing machines and the λ-calculus. A function 'interacts' with its argument to produce a result:

$$f(a) \longrightarrow \text{result}$$

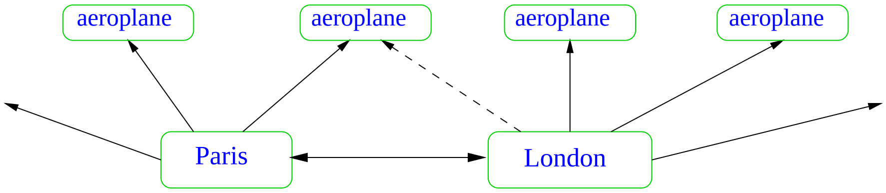
A more complicated interaction occurs when a function interacts with the state space:

$$(f(a), \text{state}) \longrightarrow (\text{result}, \text{state}')$$

Different initial states lead to different results.

The programming language ML is based on the λ-calculus. It has a formal description of the behaviour of programs, as well as a formal description of their syntax. The tool AnnoDomini, used by IBM to find Year 2000 problems in Cobol programs, is based on theoretical results underpinning ML.

A different type of interaction involving mobile agents is illustrated by the transfer of control of aeroplanes between Charles de Gaulle and Heathrow.



In this example, it is not so much results that matter, but rather the ongoing behaviour of aeroplanes moving between the two airports. Such interaction requires a different technique for modelling its behaviour. It is modelled in the π-calculus by transitions of the form

$$\text{agent} \xrightarrow{\text{E}} \text{agent}'$$

where the agent aeroplane(paris) gets information E from the environment about how to switch to London, and evolves to become aeroplane(london).

A precursor of the π-calculus was used in the construction of the air-traffic information system at Heathrow. There is much current activity in applying techniques from the π-calculus to analyse the behaviour of security protocols and distributed systems.

Natural language processing (Steve Pulman)

English and Logic

Logic developed from the study of reasoning and the notion of valid inference. Most everyday reasoning is done in a natural language like English: formal logics were developed in part in order to avoid the perceived vaguenesses and ambiguity of ordinary English.

In this project we are trying to develop simple 'reasoning agents' which can carry out various types of inference, and, more important, describe them in a manner which is transparent even to someone who is not a logician.

The demonstration shows an initial small prototype of a system which can solve syllogisms, or similar types of inference task, like Lewis Carroll's logic puzzles:

Given:

- All honest industrious men are healthy.
- No grocers are healthy.
- All industrious grocers are honest.
- All cyclists are industrious.
- All unhealthy cyclists are dishonest.

Show that:

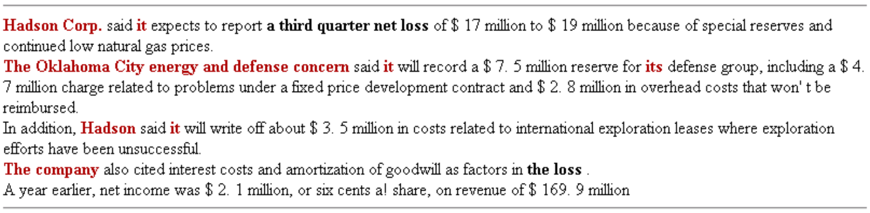
- No grocer is a cyclist

The problem is posed in simple English sentences, which are automatically translated into expressions of a formal logic. These expressions are then handed to a theorem prover which carries out the reasoning. The proof is then translated back into a sequence of simple English sentences which provide a transparent description of the reasoning involved.

The HIGHLIGHT information extraction system

Information retrieval techniques can locate documents that are relevant to our query. But how can we extract the relevant part of the document to form an answer to our question?

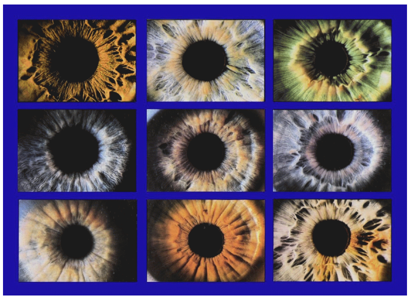
Information "extraction" uses grammatical and semantic analysis techniques to locate the relevant phrases and filter out the irrelevant ones, even when these may contain the right keywords. This analysis enables us to produce a structured representation of the relevant components of the document in a form suitable for further processing (e.g. data mining, visualisation, etc.)



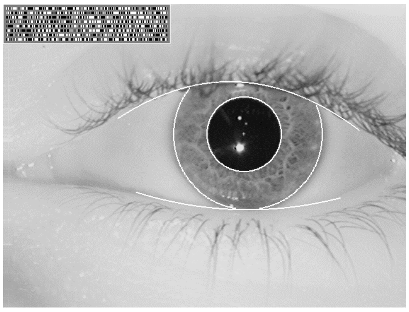
Hadson Corp. said **it** expects to report **a third quarter net loss** of $ 17 million to $ 19 million because of special reserves and continued low natural gas prices.
The Oklahoma City energy and defense concern said **it** will record a $ 7. 5 million reserve for **its** defense group, including a $ 4. 7 million charge related to problems under a fixed price development contract and $ 2. 8 million in overhead costs that won' t be reimbursed.
In addition, **Hadson** said **it** will write off about $ 3. 5 million in costs related to international exploration leases where exploration efforts have been unsuccessful.
The company also cited interest costs and amortization of goodwill as factors in **the loss** .
A year earlier, net income was $ 2. 1 million, or six cents a! share, on revenue of $ 169. 9 million

The WWW-based HIGHLIGHT system is being developed by SRI International in Cambridge. This system is partly based on work carried out in an earlier collaborative project between the Computer Laboratory and SRI.

Iris recognition (John Daugman)

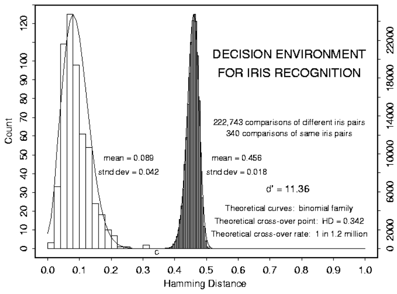


This project illustrates computer vision and pattern recognition. The purpose is real-time, high confidence recognition of a person's identity by mathematical analysis of the random patterns that are visible within the iris of an eye from some distance. Because the iris is a protected internal organ of the eye whose random texture is stable throughout life, it can serve as a kind of living passport or password that one need not remember but one always carries along.



After localizing the eyes in a face and determining the inner and outer boundaries of the irises and eyelids, the patterned texture of an iris is encoded. The mathematical process uses complex-valued 2D wavelets to extract phase information by demodulation. The result is a compact "IrisCode", whose statistical randomness spans about 266 independent degrees-of-freedom. This large stochastic dimensionality ensures that no two eyes can generate the same (or even nearly the same) IrisCode, with more than infinitesimal probability.

Because the iris pattern is a phenotypic rather than genotypic feature, even identical twins have IrisCodes that are as uncorrelated as those of unrelated people. (The same is true for comparisons of right and left eyes of one person; these again are genetically equivalent.) Therefore iris recognition outperforms even DNA testing, whose False Match rate cannot be lower than the birth rate of identical twins.



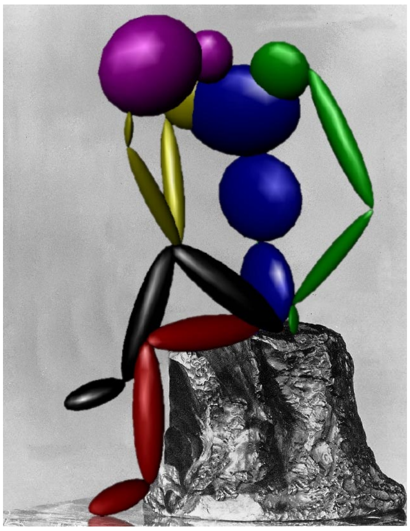
A shown by the separation in the "Decision Environment" histograms, recognition decisions have mathematical confidence level sufficient to support exhaustive searches through national-sized databases with almost no chance of a False Match. In addition to the current trials at bank cashpoint machines, iris recognition could be used in any transaction in which a person's identity must be established or confirmed. Examples are passport control, electronic commerce, benefits payment, building entry, access to privileged information, licenses, forensic or police uses, computer login, or any transaction in which personal identification currently relies on keys, cards, documents, passwords or PINs.

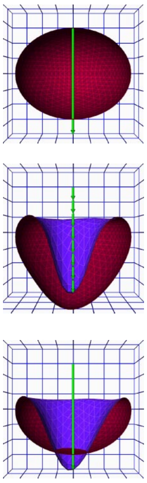
Modelling and animation (Neil Dodgson & Peter Robinson)

The Computer Laboratory has a long history of involvement in modelling for 3D computer graphics. We are currently undertaking a variety of research projects in this area.

Animation of human figures (Tony Polichroniadis & Marco Gillies)

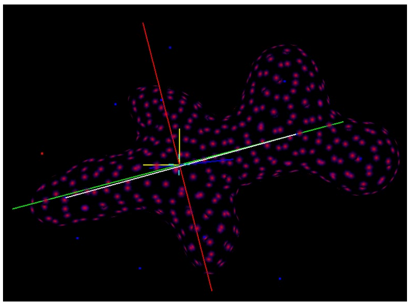
While 3D animation of insects, toys, and spaceships is now viable; modelling and animation of humans is still a research problem. We are investigating two aspects of this: how human emotion can be represented and incorporated in animation, and whether human visual psychology can be used to generate better animation algorithms.

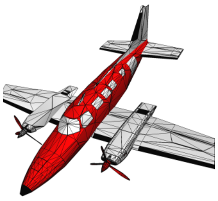


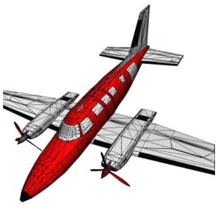


Modelling non-rigid materials (James Gain & Michael Blain)

Soft, deformable objects and liquids are extremely tricky to model effectively. We are currently tackling the mathematical underpinnings of directly manipulated free-form deformation (DMFFD) with the intention of using it for sculpting soft objects, in the same way that an artist would manipulate clay. We have also started investigations into the efficient modelling of liquids for real-time animation, particularly the mechanics of splashing.







Automatic model simplification (Jon Sewell & Peter Brown)

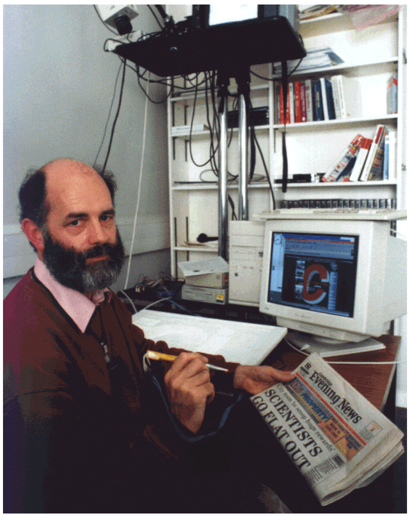
3D models can contain enormous detail. This detail is normally drawn by the computer, regardless of whether or not it is visible to the human viewer. Our work has investigated methods for automatically reducing the detail in objects without affecting the visual result. This allows 3D drawing to proceed more rapidly and can be applied in flight simulators and other complex virtual environments

Mathematical foundations (Brett Saunders & James Gain)

Complex mathematics lie behind seemingly simple models. The non-rigid materials work (above) tackles some fierce maths. Another current project investigates methods of effectively simplifying the mathematics of constrained dynamic animation, to allow for better real-time effects. In all this work we endeavour to keep a balance between the completeness of our models of the world and their computational tractability.

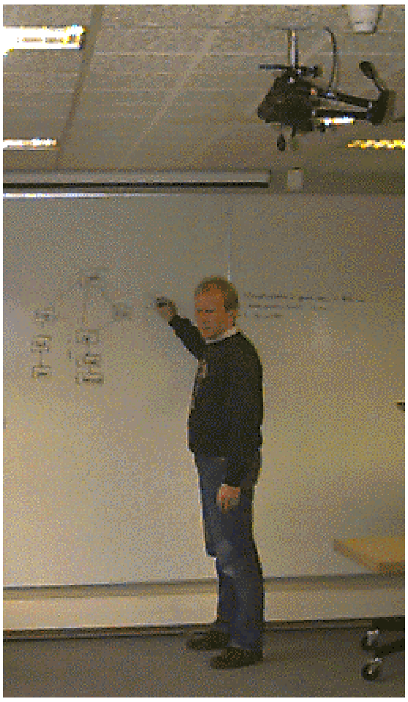
Video user interfaces (Peter Robinson & Richard Watts)

As part of our work on human-computer interaction, we are looking at ways of using digitised video from television cameras in user interfaces for computers.



DigitalDesk

The *DigitalDesk* is built around an ordinary desk and can be used as such, but it has extra capabilities. A video camera is mounted above the desk, to detect where the user is pointing and to read documents. Computer images are also projected down onto the work surface and onto real paper documents.

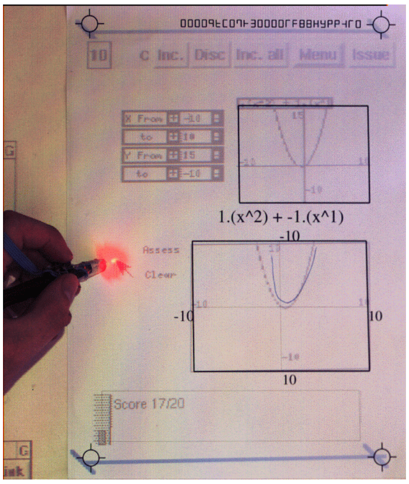


BrightBoard

Another approach is used in *BrightBoard*. A video camera is pointed at an ordinary whiteboard and its output fed into a computer. The whiteboard thus becomes an alternative way of controlling the computer.

Animated paper documents

This technology is being applied to electronic publishing. We have combined electronic and printed documents to give a richer presentation than that afforded by either separate medium. Material is published as an ordinary, printed document that can be read in the normal way, enjoying the usual benefits of readability, accessibility and portability. However, when observed by a camera connected to a computer, the material acquires the properties of an electronic document, blurring the distinction between the two modes of operation.



These systems show how computers can be built into everyday objects with simple user interfaces that do not require expert knowledge to operate.

Computing Service exhibition

Overseer: Richard Stibbs

The half-century since May 1949 has been remarkable not only in terms of the academic and technical achievements of the Maths Lab but also in the close and symbiotic relationship between Teaching & Research and the Computing Service. This has been a tribute to successive Heads of the Department – Maurice Wilkes, Roger Needham, and Robin Milner – and to Directors of the Computing Service – David Hartley and Mike Sayers.

The formal institution of the Computing Service in the early 1970s as a division of the Computer Laboratory rather than as a separate department went against the orthodox thinking in the universities of that time, but that decision has been fully justified. There have been the advantages of the shared services – the printroom, accounts, building services, stores, reception, telephonist, the common room, and the library – which split into two departments would have been more costly. There have also been many other less tangible benefits. Computer Service staff continue to lecture for the Computer Science Tripos (CST) and Diploma, and many CS staff have served as Directors of Studies and as supervisors. The Computing Service has been able to liaise closely with T&R in the provision of hardware and software for the CST, and has worked to ensure the success of T&R service teaching commitments (especially for the Natural Sciences Tripos, Part IA). Exposure to T&R research projects has given CS staff early introductions to new technologies (two examples of which are the Internet Protocols and Ethernet technology) and the informed and constructive criticism from T&R of CS systems continues to be a helpful spur to excellence. Many generations of T&R research students were exposed to practical aspects of their science by being employed as Programming Advisers (the precursor of the Help Desk) and as demonstrators for CS courses. The members of the engineering staff of the Computer Laboratory who started the period providing services to research projects have developed their skills within the Laboratory to provide the highly regarded hardware maintenance service to the University.

This tone of co-operation outside of research was initiated by Maurice Wilkes and exemplified by his willingness to share the Lab's resources with the rest of the University and by the decision to build a viewing gallery for the Titan room.

The institution of the Computing Service in 1970/1 could have been the beginning of a parting of the ways but David Hartley and his colleagues stayed close to the ethos of usability and efficiency which had developed during the development of the Titan Operating System and it was with T&R encouragement and support that the Service decided to embark on the technically difficult but very rewarding task of adapting commercial systems to provide the ease of use to which the University had become accustomed in the Titan era. The result was the Phoenix system which provided what was widely accepted to be the best main-frame academic service in the world. T&R staff were very involved users of the Service and provided both valuable criticism and staffing for the Advisory Services and in many cases joined the Computing Service staff. The shared common room and the proximity of Reception and the User Area to T&R certainly helped as did the chatting over the output tanks.

The excellence of the Phoenix system meant that the University had no need to pass through the rather painful phase of CPM-based machines and it was the inspired development of the BBC micro in the Lab that gave our users the luxury of dual-use machines providing both terminal emulation and stand alone use. PCs and Macs followed to take the place of the Beebs and the increase in local compute power led to the need to introduce the PWF (Personal Workstation Facility running Novell Netware) to administer the distributed computing.

Meanwhile the Service had been developing the underlying networking infrastructure, taking part in local and national standardisation on the Coloured Book protocols, and later adopting the Internet protocols. David Hartley, Mike Sayers and Roger Needham had seen the need to develop the University's physical networking and obtained the late 80s the achievement of the agreement between the Colleges and the University to install the 33km of ducting containing optical fibre cables under Cambridge making up the Granta Backbone network. This allowed the rapid development of distributed computing around the University to today's situation where we support 18,000 Ethernet connections and 28,000 registered users.

In parallel, there have been developed the Central Unix Service, Web, News, Dial-up and Mail Servers and the many and varied support services including the Literary and Linguistics Computer Centre, the Photography and Illustration Service, the Help Desk, Technical User Services, Unix Support, the Printroom, the Information Service, User Administration, Software Sales, Reception, Video-Conferencing, Hardware Maintenance and the increasingly important local Computer Security team fending off world-wide hacker attacks.

However a golden anniversary must be a time to be nostalgic and although we can now deliver unprecedented power and connectivity to desktops throughout the University there is no longer the sense of the excitement of the shared late night endeavours on a single system that can remembered from the Maths Lab and the various CS User Areas. We hope the display of photographs in the Austin stairwell will be a happy reminder of that shared excitement.

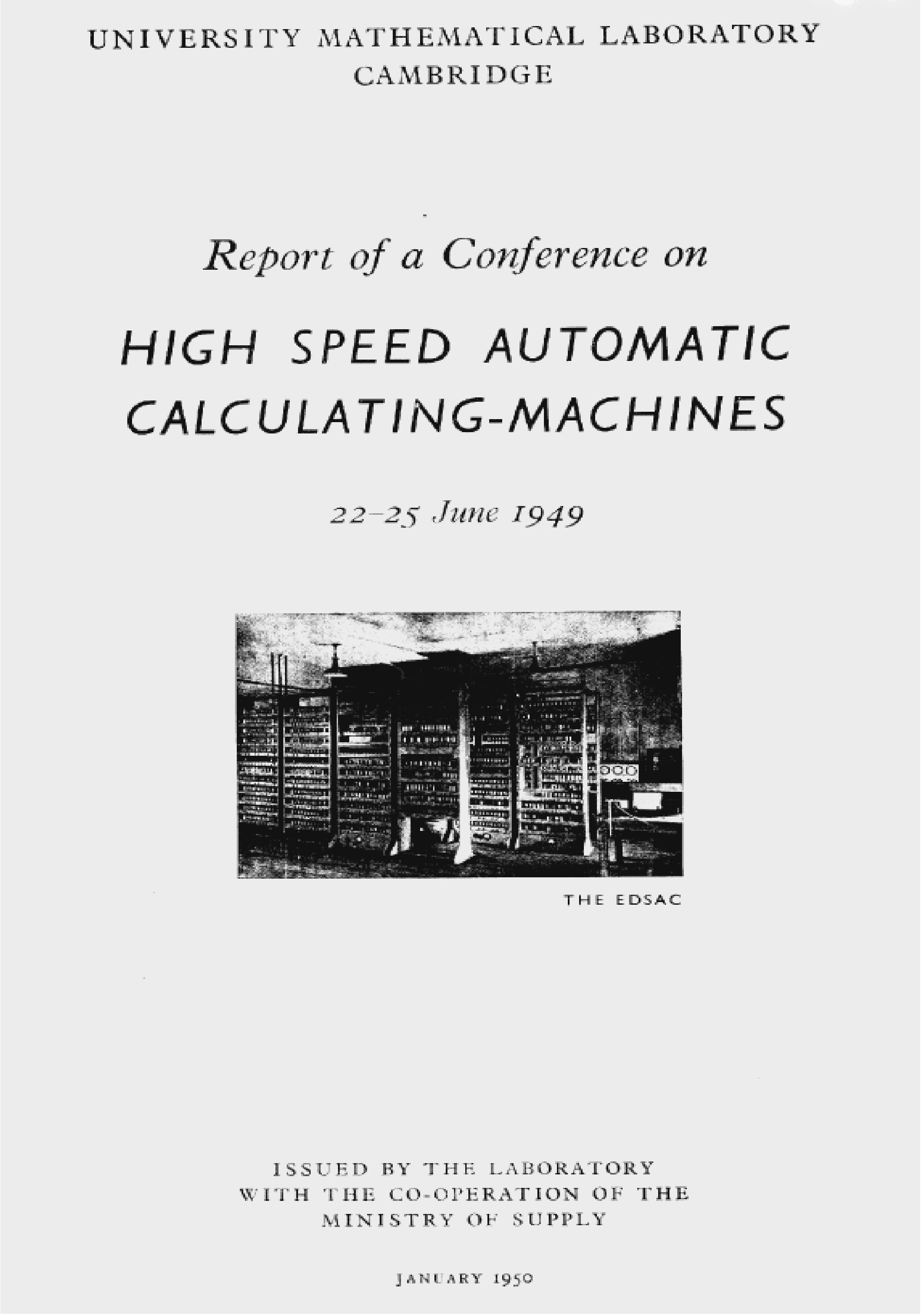
New Computer Laboratory Building plans

The old Laboratory from the courtyard

The Mathematical Laboratory building between the New Museums Site and Corn Exchange Street, in which the EDSAC was constructed and used, was a 19th Century building previously occupied by the Department of Anatomy. This building, which also housed EDSAC 2 and Titan, was replaced during the 1960s by a hefty new building shared between several departments, into which the Laboratory moved in 1969. In the thirty years since then, with the growth of research, teaching, and the computing service, the Computer Laboratory has spread out over adjoining buildings on the New Museums Site and to offsite outstations. Its current state is inconvenient and constrained. A new, more commodious and practical building is planned for the University's West Cambridge Site, for which a very substantial gift from the William H. Gates III Foundation has been promised. Detailed design work on the new building is now underway, and current draft plans will be on show.

Extracts from the 1949 conference

A conference on *High speed automatic calculating-machines* was held in Cambridge in June 1949 This was a major event with 35 presentations and discussions, and attracted 144 delegates from the USA, France, Germany, Holland and Sweden as well as the UK. The opening presentation about the EDSAC by M.V. Wilkes and W. Renwick is reprinted here together with Miss B.H. Worsley's summary of a demonstration of the EDSAC given by W. Renwick.



UNIVERSITY MATHEMATICAL LABORATORY
CAMBRIDGE

Report of a Conference on

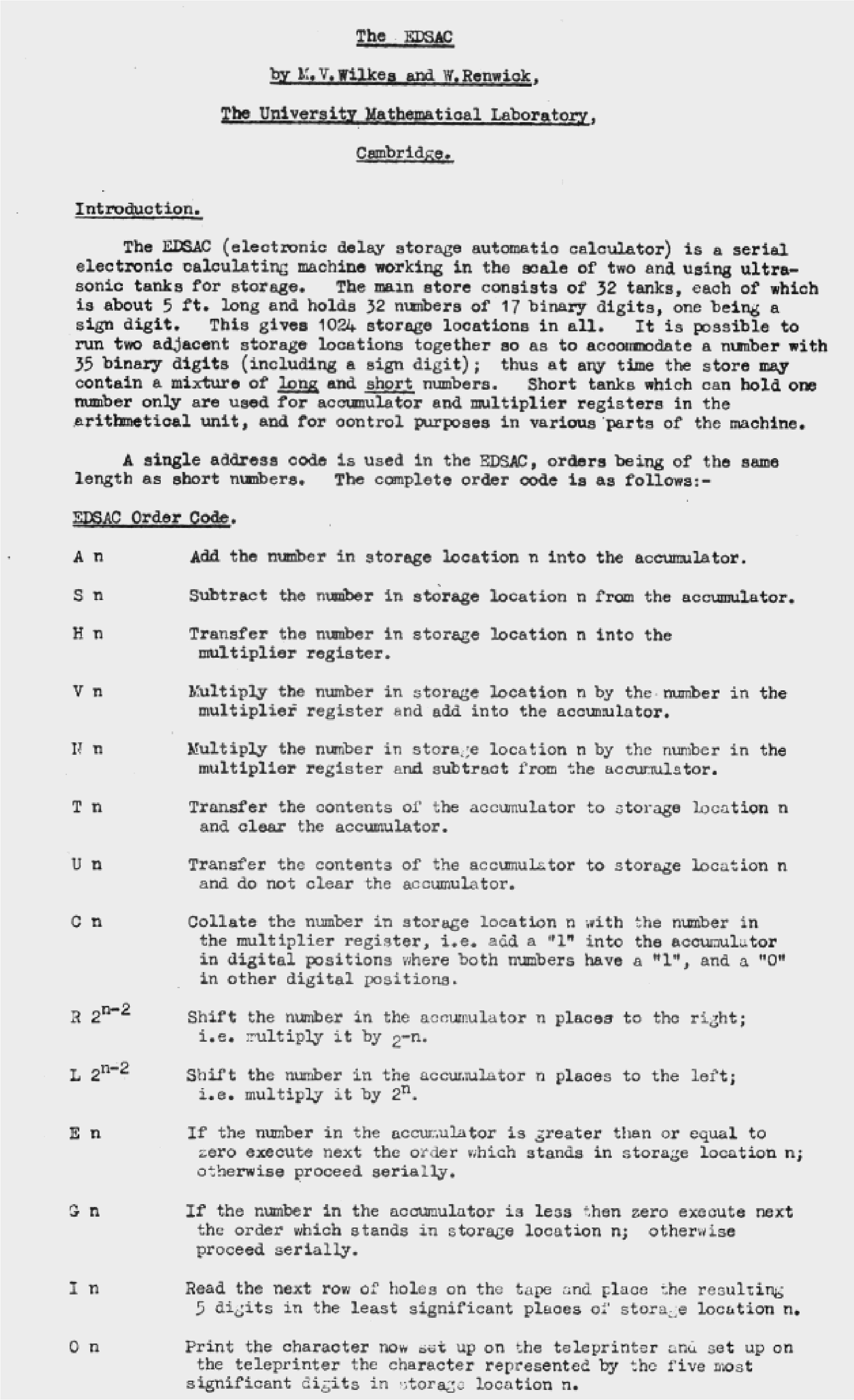
HIGH SPEED AUTOMATIC
CALCULATING-MACHINES

22–25 June 1949

THE EDSAC

ISSUED BY THE LABORATORY
WITH THE CO-OPERATION OF THE
MINISTRY OF SUPPLY

JANUARY 1950



The EDSAC

by M.V.Wilkes and W.Renwick,

The University Mathematical Laboratory,

Cambridge.

Introduction.

The EDSAC (electronic delay storage automatic calculator) is a serial electronic calculating machine working in the scale of two and using ultrasonic tanks for storage. The main store consists of 32 tanks, each of which is about 5 ft. long and holds 32 numbers of 17 binary digits, one being a sign digit. This gives 1024 storage locations in all. It is possible to run two adjacent storage locations together so as to accommodate a number with 35 binary digits (including a sign digit); thus at any time the store may contain a mixture of <u>long</u> and <u>short</u> numbers. Short tanks which can hold one number only are used for accumulator and multiplier registers in the arithmetical unit, and for control purposes in various parts of the machine.

A single address code is used in the EDSAC, orders being of the same length as short numbers. The complete order code is as follows:-

EDSAC Order Code.

A n	Add the number in storage location n into the accumulator.
S n	Subtract the number in storage location n from the accumulator.
H n	Transfer the number in storage location n into the multiplier register.
V n	Multiply the number in storage location n by the number in the multiplier register and add into the accumulator.
N n	Multiply the number in storage location n by the number in the multiplier register and subtract from the accumulator.
T n	Transfer the contents of the accumulator to storage location n and clear the accumulator.
U n	Transfer the contents of the accumulator to storage location n and do not clear the accumulator.
C n	Collate the number in storage location n with the number in the multiplier register, i.e. add a "1" into the accumulator in digital positions where both numbers have a "1", and a "0" in other digital positions.
R 2^{n-2}	Shift the number in the accumulator n places to the right; i.e. multiply it by 2^{-n}.
L 2^{n-2}	Shift the number in the accumulator n places to the left; i.e. multiply it by 2^{n}.
E n	If the number in the accumulator is greater than or equal to zero execute next the order which stands in storage location n; otherwise proceed serially.
G n	If the number in the accumulator is less then zero execute next the order which stands in storage location n; otherwise proceed serially.
I n	Read the next row of holes on the tape and place the resulting 5 digits in the least significant places of storage location n.
O n	Print the character now set up on the teleprinter and set up on the teleprinter the character represented by the five most significant digits in storage location n.

F n	Place the five digits which represent the character next to be printed by the teleprinter in the five most significant places of storage location n.
X	Round off the number in the accumulator to 16 binary digits.
Y	" " " " " " " " 34 " "
Z	Stop the machine and ring the warning bell.

It will be seen that most orders consist of a functional part which defines the operation, and a numerical part which defines a storage location; some orders, however, consist of a functional part only.

Ordinary 5-hole punched tape of the kind used in telegraphy is used for input. Each row of holes represents a 5-digit binary number and the basic input operation is to transfer this number to the store. Similarly, the output mechanism is a teleprinter, and the basic output operation is to transfer a 5-digit binary number to the printer, and to print the corresponding character. The teleprinter code is chosen so that binary numbers up to nine are printed as the corresponding figures, and a similar code is used for input. This enables the operation of conversion to and from the decimal system to be programmed as part of the calculation.

The purpose of the F order is to enable the operation of the printer to be checked. Apart from this no special checking facilities are provided in the EDSAC, it being left to the programmer to incorporate in the programme such checks as he considers necessary.

The control sequence.

When the machine is in operation, orders are executed automatically in the order in which they stand in the store. The only exception is when a conditional order (E or G) is encountered, and the condition is satisfied; the next order to be executed is then the one which stands in the storage location specified in the conditional order.

Count is kept of the orders as they are executed by means of a short tank - known as the sequence control tank-which has associated with it an adding circuit through which unity is added to the number stored in the tank each time an order is executed. During a conditional order (when the condition is satisfied) unity is not added, but instead the number in the tank is replaced by the numerical part of the conditional order.

The control sequence of the machine falls into two parts. In stage I, an order is transferred from the location in the store given by the number in the sequence control tank to a short tank known as the order tank. In stage II the order in the order tank is executed.

The various constituent units of the machine store, arithmetical unit, input unit, output unit, sequence control tank, and order tank - are connected together through gates, so that the interconnections proper to each successive part of the control sequence can be made by opening the appropriate gates. The gates are operated by waveforms supplied either by the main control unit, or by a part of the machine given, for the purpose of this description, the name "order interpreter." In the machine itself the "order interpreter" consists of a number of separate units. In the diagrams that follow, wires which carry pulses are shown as continuous lines, while those which carry control waveforms are shown dotted.

Figure 1 shows the state of the machine during Stage I of the control sequence, it being assumed that the storage location specified by the number in the sequence control tank is in tank 1 of the main store. The wire along which the order passes from this storage location to the order tank is shown by a heavy line in the diagram, and similarly the wires along which the control waveforms pass are shown as heavy dotted lines. The location section of the "order interpreter" receives the number in the sequence control tank, and proceeds to emit a waveform which opens the output gate of tank 1. There are 32 orders (or short numbers, or a mixture of orders, short numbers and long numbers) circulating in the tank, and this gate is opened as soon as the required number becomes available and closed again when it has passed through. In this it is treated differently from the other gates in the machine which remain open or closed during the whole of stage I; the difference is indicated in the diagram by the different form of shading used. The switching and timing necessary for the operations detailed above are carried out within the "order interpreter."

Stage II is very similar to stage I except that the type of operation performed is not restricted to a simple transfer from the store to the order tank. During stage II the order tank is connected to the "order interpreter" in place of the sequence control tank, and the numerical part of the order is dealt with in the same way as the number in the sequence control tank during stage I. A new feature, however, is that the function section of the "order interpreter" comes into action, and interprets the functional part of the order; it proceeds to emit waveforms which (a) set up connections between the units ready for the passage of numbers and (b) place the arithmetical unit (or input or output units as the case may be) in a state of readiness for executing the order. Figure 2 shows the various gates set up for the execution of an add order calling for the transfer of a number from tank 2 of the main store to the arithmetical unit. One of the control wires passing from the "order interpreter" to the arithmetical unit is shown as a heavy (dotted) line to indicate that the arithmetical unit has been prepared for some operation such as addition.

The above description has been written with the case of an arithmetical order involving the store in mind. In the case of other orders (e.g. left or right shift, or a conditional order) stage II is simplified in that the location section of the "order interpreter" is not used. For example, during stage II for a conditional order, the order tank is connected to the sequence control tank through the gate shown in the diagram and the whole order transferred from one tank to the other. When it is in the sequence control tank, only the numerical part is used, the function part being irrelevant.

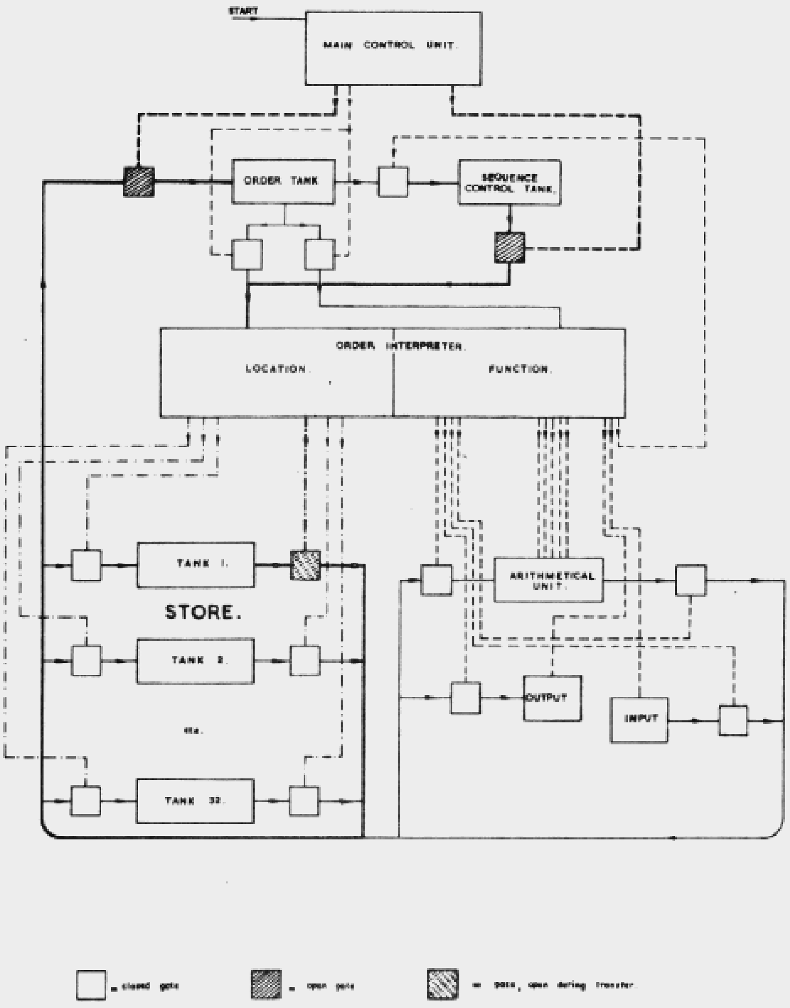
Initial orders.

From what has been said, and from an examination of the order code, it will be seen that the input mechanism is controlled by programme orders. Unless, therefore, there are some orders in the store to begin with, nothing can be taken in through the input, and the machine cannot start. For this reason there is a sequence of orders - known as initial orders - permanently wired onto a set of uniselectors (rotary telephone switches). These orders can be transferred to the store by pressing a button.

There is considerable latitude in the choice of the initial orders, although once they have been wired onto the uniselectors it is not easy to change them. The initial orders at present used in the EDSAC enable orders punched in the following form to be taken in from the tape. First a letter indicating the function is punched, then the numerical part of the order, in decimal form, and finally the letter L or S indicating, respectively, that the order refers to a long or a short number. If the order has no numerical part it is punched simply as a letter followed by S. Under the control of the initial orders the machine converts the numerical part of the order to binary form, and assembles the order with the function digits and the numerical digits in their correct relative positions.

References.

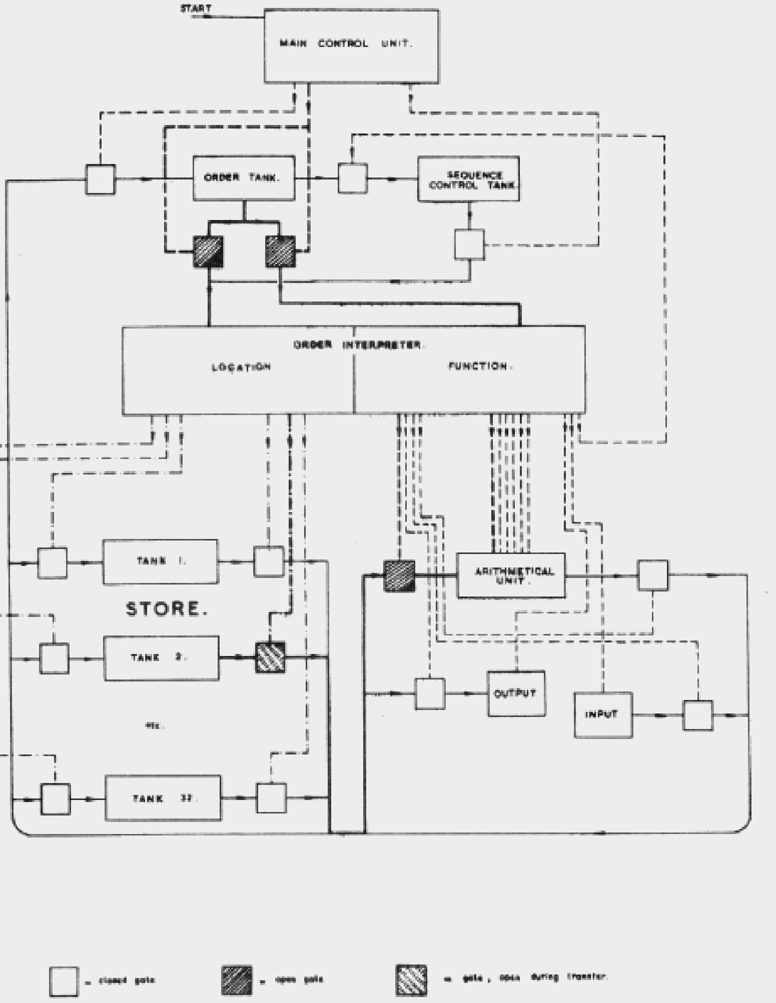
1.	Wilkes, M.V.	Proceedings of the Royal Society 195, 274, (1948).
2.	Wilkes, M.V.	Journal of Scientific Instruments 26, 217, (1949).
3.	Wilkes, M.V. and Renwick, W.	Electronic Engineering 20, 208, (1948).



STAGE I.

FIG I.

E1/160/1. | DATE



STAGE 2.

FIG 2.

	DATE
E1/170/1.	?

THE E.D.S.A.C. DEMONSTRATION (Mr. W. Renwick)

Account prepared by

Miss B.H. Worsley.

During the first day of the Conference, a demonstration was given of the new Cambridge Calculator, the E.D. S.A.C. In this demonstration, tables of both squares and primes were printed. These two problems have served as test routines for the machine, and are detailed here for the record rather than as examples of elegant programming. No attempt will be made to explain all the tricks and devices available to the programmer. However, it is hoped that the following notes, flow diagrams, and annotations will render the actual routines used in the demonstration intelligible.

Description of the Demonstration.

The following sequence of events was observed:-

1. The teleprinter tape, punched with the orders (ii) or (iii) (see below) was put in place on the tape reader, and the start button was pressed. The remainder of the operation then proceeded without further intervention on the part of the operator.

2. The initial orders (i) (see below) were then input directly into the first 31 short store positions through the action of the uniselectors. These orders, used in every problem, are pre-wired in their binary coded form on the uniselectors, from which they are transferred automatically to their appropriate store locations. This operation took about 5 seconds, and was accompanied by a series of clicking sounds at the rate of about 6 a second.

3. The tape-reader was next brought into action so that the orders (ii) or (iii) were synthesized to their binary-coded form, and placed in consecutive store positions starting from position 31. This action was under the control of the initial orders (i), thus utilizing the machine itself to effect the necessary conversion. Input was at the rate of one symbol per 150 milli-seconds, or, on the average, $1\frac{1}{2}$ orders per second, one click being heard for each symbol.

4. Finally, the teleprinter was seen to begin operation, the computation and output then being under the control of the internally stored orders (ii) or (iii). The time between the output of consecutive primes became appreciable as the numbers tested became larger. The printing of the squares was about as fast as the typing action, namely, one symbol per 150 milli-seconds.

Throughout the entire operation, the contents of the following tanks were displayed on Cathode Ray Tube screens in the binary form in which they are used in the machine:

1. The 16 words (32 short numbers or orders) in any one of the long storage tanks desired.
2. The Accumulator.
3. The counter tank.
4. The multiplier.
5. The multiplicand.
6. The order sequence tank.

A brief informal history of the Computer Laboratory

Compiled by Karen Spärck Jones using material from the University Reporter; and from Maurice Wilkes, David Wheeler, Roger Needham, David Hartley, and Brian Westwood, to all of whom the compiler is grateful for help and comments.

This note indicates some of the more memorable things in the history of the Laboratory; it is not intended to be a full account of the Laboratory's activities, or a formal historical record or audit.

For further historical reference see:
M.V. Wilkes, *Memoirs of a Computer Pioneer*, MIT Press, 1985.
IEEE Annals of the History of Computing,
Special issue on the University of Cambridge, Vol 14 (4), 1992.

1936 General Board Report on the Establishment of a Computing Laboratory (2 December) referred to 'recent developments in mechanical and electrical aids to computation' and types of machine:
'great use is now made of them in all branches of science'
'The important feature of these machines ... is the speed with which definite answers can be produced.'

Laboratory intended (cf later GB Report 1969) 'to provide a computing service for general use, and to be a centre for the development of computational techniques in the University'.

1937 14 May
Founded as Mathematical Laboratory, Director Professor J.E. Lennard-Jones (Professor of Theoretical Chemistry), only staff member M.V. Wilkes, appointed as University Demonstrator.

To be used for mechanical computation with a differential analyser, desk machines, etc.

The Laboratory was to be housed in the North Wing of the former Anatomy School, on the New Museums Site.

Not yet opened when taken over by Ministry of Supply for the war. Wilkes already working elsewhere on radar and, later, on operational research.

1945 Laboratory returned to civilian use, with Wilkes (temporary University Lecturer) as Acting Director.

1946 General Board Report on the Organisation of the Laboratory (24 July):
'the Laboratory has been equipped with a number of the most modern calculating machines, ... and library and workshop facilities have been built up and assistants appointed.'
'its services will be available to all departments of the University'
'[Dr Wilkes] has formed projects of research work ... which will be of great value to the science of computation.'

Wilkes appointed Director with 'the duty to advance knowledge of the science of mathematical computation, to promote and direct research in it, and to supervise the work of the ... Laboratory under the general control of the Mathematical Laboratory Committee.'

1946 October
Work began on the EDSAC (Electronic Delay Storage Automatic Calculator, later referred to as EDSAC 1), following Wilkes' visit to the US in August and September.

1947 Messrs J. Lyons and Company gave a grant of 3000 pounds for research in the Laboratory, with no attached conditions, and support for an assistant for a year (in practice much longer).

Student volunteers, V. Hale, B. Haselgrove and D.J. Wheeler, helping with constructional work.

1947-8 Thursday Colloquia began; first talk B. Noble on programming for the EDSAC.

Laboratory's first research student, J.M. Bennett, followed by D.J. Wheeler.

Laboratory members working under Wilkes and all contributing in some way or other to the EDSAC project included academic staff: W. Renwick, B. Noble, D.W. Willis and E.N. Mutch; assistant staff: P.J. Farmer, G.J. Stevens, S.A. Barton, R.S. Piggott, L.J. Foreman and P. Chamberlain; and research students: Bennett and Wheeler.

(Laboratory assistant staff list for 1948 has 14 people, including two unestablished 'boys' and a part-time cleaner; photograph of 1948 has 19 people.)

1948-9 Research on programming methods under Wilkes, including: definition and refinement of Initial Orders (Wheeler); closed subroutines (Wheeler); building of a library of subroutines (all laboratory members interested in programming, plus Professor D. R. Hartree).

1949 6 May
First logged program on EDSAC 1 (computing squares of 0-99).
This was the first complete and fully operational regular electronic digital stored program computer; Manchester's absolute first, in 1948, was the Small Scale Experimental Machine, built to validate innovative CRT memory technology. (These machines were before the first US machines.)

Normal operation, with paper tape input, for range of user calculations. The Initial Orders, (a primitive assembler) were hard wired on to rotary telephone switches.

June
Conference on high-speed automatic calculating machines, first outside US, with 100 participants; EDSAC 1 demonstrated and came out very well. (Report published 1950, later version, edited by M.R. Williams and M. Campbell-Kelly, in MIT/Tomash historical series, 1989.)

General Board report on the Organisation of the Laboratory (19 July):
'the Laboratory has developed until it now occupies a leading position among the mathematical laboratories of the country.'
It has 'calculating machines ... Hollerith equipment ... and a high speed electronic calculating machine has been designed and largely constructed in the Laboratory. The work on this machine has already won recognition and financial support for the Laboratory from outside sources.'

Laboratory therefore freed from its Committee; Wilkes became Head of Department and Renwick (chief engineer) a University Demonstrator.

1950 Program service began on semi-formal basis, with an operator to run programs (except at night, when authorised users were entitled to run the machine till it broke, a tradition continued with EDSAC 2); early operators included E. Breakwell (in the 1951 film), V. Webber, R. Hill.

Significant development of library routines, e.g. S. Gill's Runge Kutta program, Wheeler's interpretive floating point routine. Interpreter by Bennett to allow compact programs. First steps in computation for X-ray crystallography also by Bennett.

First Summer School on Programme Design for Automatic Digital Computing Machines, with 51 attendees (they ran till 1958).

Further research students working in the Laboratory (4 listed in 1950) included Gill, A.S. Douglas, B. Worsley, E.S. Page.

University staff 4: Wilkes, Renwick, with R.A. Brooker and Mutch Assistants in Research.

Others outside the Laboratory also involved with EDSAC 1, notably Professor Hartree.

1951 Film made of EDSAC 1.

'The preparation of programs for an electronic digital computer' by Wilkes, Wheeler, and Gill, Addison-Westley, the first book on programming to be published.

LEO computer, world's first business computer, developed by Lyons and based on EDSAC 1, began operation.

Wilkes' paper, 'On the best way to design an automatic calculating machine', introduced the idea of microprogramming and bit-slicing.

Work began on EDSAC 2, with a grant from Nuffield Foundation.

1952 Magnetic tapes introduced experimentally for EDSAC 1.

1952 onwards
Continuing development of programs and methods e.g. hash tables, recursion with stack, program labels (Wilkes), Fourier transforms.

1953 Diploma in Numerical Analysis and Automatic Computing began ('automatic computing' then best shot at 'computer science');
one-year postgraduate course, the first formal course leading to a university qualification anywhere in the world. Motivated by Mathematics Faculty Board Report on the 'demand for postgraduate instruction in numerical analysis and automatic computing ... [which] if not met, there is a danger that the application to scientific research of the machines now being built will be hampered'.
The Laboratory 'was one of the pioneers in the development and use of electronic computing-machines (sic)'.
'The Summer School deals [only] with "programming", rather than the general theory of the numerical methods which are programmed.'
The Diploma 'would include theoretical and practical work ... [and also]

instruction about the various types of computing-machine ... and the principles of design on which they are based.'
With only a few students initially, no extra staff would be needed.

University-supported teaching and research staff in the Laboratory were Wilkes, J.C.P. Miller, Renwick, Mutch, and Gill, joined slightly later by C.B. Haselgrove.

1953 onwards
Heavy use of EDSAC 1 for (among other subjects) theoretical chemistry under S.F. Boys; X-ray molecular biology by J.C. Kendrew; numerical analysis by Hartree, Miller; atmospheric oscillations by Wilkes; early work on radioastronomy (much extended on EDSAC 2) by group under M.F. Ryle.

Priorities Committee established for approval of computing projects and resource allocation.

1954 Diploma 2 written papers and 1 3-hour practical (4 hours 1955). 3 successful candidates.

1956 As many as 100 attended the Summer School.

1956-7 EDSAC 1.5, (EDSAC 2 with Wheeler's small control matrix, precursor of EDSAC 2), running and used by J. Blackler (later J. Wheeler) for astrophysics.

1958 EDSAC 2 taking load from EDSAC 1.

11 July
EDSAC 1 shut down.

EDSAC 2, the first full-scale microprogrammed machine, also the first bit-sliced machine. Used fast paper tape for I/O, magnetic tapes. A small number of basic library subroutines were wired into a ROM.

1960 University-supported academic staff now 6, Wilkes, Wheeler, Miller, H.P.F. Swinnerton-Dyer, Mutch, and Mrs M.O. Mutch. Total staff estimated about 30 including about 10 engineers.
About 7 PhD students and 10 Diploma students.
Number of users unknown, but at least 50 (a lot for those days).

1961 Autocode for programming, developed by D.F. Hartley, following stimulus from Manchester, in service.

Business game, implemented by J. Hillmore (a Diploma student); attractive application with many takers, e.g. Her Majesty's Treasury.

Proposal for Titan, modified version of Atlas then being developed by Ferranti. W. S. Elliott joined the Lab as Senior Project Engineer. Joint project with Ferranti and successor companies (ICT and ICL); some automated hardware design done at Cambridge; operating system developed jointly.

Designed originally for multi-processing; design modified after Wilkes' visit to MIT in 1963 to support multiple-access (provision for up to 64 terminals).

Had slave store – nowadays instruction cache (Wheeler); was also designed for very rapid response to interrupts, with more operating system scheduling than was then usual.

N.E. Wiseman became Chief Engineer.

Work on design of CPL, a higher-level language intended for Titan, initiated by Hartley and D.W. Barron; C. Strachey later joined the project, and the then University of London Institute for Computer Science became involved. A seminal project.

1962 Large (16 K words) memory addition to EDSAC 2.

1964 Titan came into service with first operating system (Temporary Supervisor by Swinnerton-Dyer). Magnetic tapes.

1965 EDSAC 2 switched off.

Autocode, first higher-level language on Titan. Followed later, under pressure from scientific users, by FORTRAN.

Fifteen years of CAD research, initially led by C.A. Lang, began, using a PDP-7 and DEC Type 340 display (the first outside the USA) connected by data-link to the Titan. It later involved a highly innovative numerically-controlled machine for cutting models of metal parts in plastic foam.

Diploma renamed Diploma in Computer Science.

1966 Titan Main Supervisor replaced the Temporary Supervisor.

1967 Titan multiple-access system on a 24-hour, 7-days-a-week basis to users outside the Laboratory (film of system made in 1968). Discs had been installed to support this.

R.M. Needham originated and installed now almost universal practice of storing passwords with one-way function; also a quota system for allocating file storage.

1968 M. Richards returned from MIT and continued work on BCPL, a language with roots in CPL he had developed, and an ancestor of the still widely used C.

Vigorous growth of computing service under the Superintendent, E.N. Mutch; about 200 users of the multiple-access system.

Line of work on automated algebra began under D. Barton, carried on by J.P. Fitch, (Adams Prize 1975), S.R. Bourne and A.C. Norman.

1969 Move to new (present) building on an adjoining site. Titan airlifted by crane ('the computing service is suspended'). Service maintained for three months using another Atlas 2 at CAD Centre (Ministry of Technology).

Sad early death of E.N. Mutch, while move was in progress.

General Board Report on the Laboratory and University Computing Service (22 October):
'from the very beginning EDSAC 1 was made available to anyone in the University who could make good use of it; and such users were able to obtain advice and assistance from ... the Laboratory.'
'the steady improvement ... of Titan has enabled it to meet the steadily increasing needs of the computing service, providing in the process the first substantial multiple access system to be available in a British University.'
'Over the last twenty-five years the ... Laboratory has become a service

department comparable only with the University Library', imposing a heavy burden on the staff.

Major reorganisation followed.

Research on screen editing under Wiseman using the PDP-7.

1970 Laboratory teaching, research and technical staff 24, including externally funded (and only 1 Lecturer). Total staff 75.
About 27 PhD students, 21 Diploma.

Mathematical Laboratory renamed Computer Laboratory, became institution independent of any faculty, responsible to the Computer Syndicate.

Computing Service divided from teaching and research within Laboratory. D.F. Hartley became first Director of the University Computing Service.

1970-7 CAP project on memory protection, based on capabilities implemented in hardware, under Wilkes and Needham with Wheeler responsible for implementation (BCS Technical Award 1978 for 'CAP (Capability Protection) Project' to Needham).

1971 One-year Computer Sciences Tripos, first independent undergraduate teaching (started one year earlier as a Part II within Natural Sciences Tripos, though computing was not deemed an experimental subject), 34 students.

Diploma and Tripos each 4 exam papers.

IBM 370/165 installed for the Computing Service.

S.A. Barton became Chief Engineer after Wiseman became Lecturer.

1972 Wiseman seconded to the Cambridge University Press for the development of a computer-aided type setting and book production system; very successful.

1973 Titan switched off.

IBM memory doubled from 1Mbytes to 2, file store 1000Mbytes; PDP-11 communications front end for 10 Remote Job Entry stations, 133 terminal lines; interactive graphics unit based on a PDP-11; three-shift operation, two outstations.

Computing Service established staff 26 plus 5 engineers.
About 2500 users.

1974 Cambridge RING project initiated, involving many people;
National standard 1982.

Shape Data Ltd started, probably the Laboratory's first spin-off company.

1975 Phoenix Command Language for IBM 370/165.

1977 Initial work on University Data Network as a service, independent of the mainframe computer.

Beginning of formal project, under the Government's Advanced Computer Technology Projects, to develop a chip implementation of the Cambridge Ring. Led to research in electronic CAD (P. Robinson).

1978 Two-year Computer Science Tripos began.

Research on Cambridge Model Distributed System began.

1979 University-wide hardware maintenance service formalised and expanded.

A. Hopper appointed an Assistant Lecturer.

1980 Professor M.V. Wilkes retired. Succeeded as Head of Department by R.M. Needham.

M.J.C. Gordon joined the Laboratory from Edinburgh University and inaugurated a new line of research in formal methods.

Teaching and research staff 10. Computing Service established staff 31, plus 6 engineers. Total Laboratory staff 98. Also about 10 postdoctoral research fellows and assistants.
43 PhD students, 15 Diploma, 98 Tripos.
About 1400 active users.

Service 368 communication links, 3 outstations.

Development work on Fast Ring begun.

1980 onwards
Major research projects included formal verification of hardware designs (Gordon); Project UNIVERSE, interconnection of LANs by satellite (Needham, A.J. Herbert and I.M. Leslie); Rainbow display (Wiseman) (BCS Technical Award 1985 for 'The Cambridge Rainbow Display' to Wiseman).

[R.M. Needham : "halcyon days" – an expanding Laboratory and no external interference.]

Expansion of mass teaching in programming, led by F.H. King.

Creation of Supporters Club led by J.A. Lang, by mid-90s having several dozen companies including many with personal origins in or links with Laboratory students or staff.

Continued takeover of accommodation (laboratory came to extend from Corn Exchange Street to Free School Lane, via 2 bridges!).

1981 BCS Technical Award to the Computer Laboratory for 'The Cambridge Digital Communication Ring'.

1982 IBM 370/165 replaced by IBM 3081D.

Development of 'JNT-PAD', microcomputer network unit building block for university X.25 networks in U.K.

UNIX system on VAX 11/750 as second service within Laboratory; but also gradual transition during 80s for research side to have its own machines.

1983 Major expansion of Laboratory (teaching and research) with 5 new posts.

1985 One-year MPhil in Computer Speech and Language Processing, jointly with Engineering Department. Consolidation of research on natural language processing (K. Sparck Jones, S.G. Pulman).

Verification of VIPER chip (Gordon and A. Cohn).

Line of research on middleware began (J.M. Bacon, K. Moody).

1985 onwards
Collaboration with nearby industrial research establishments, especially SRI International and Xerox EuroPARC.

1986 Project UNISON, distributed systems (Leslie).

Link with Olivetti Research Laboratory (ORL) established.

1987 50th Anniversary of the Laboratory.

1987 onwards
Further development of research on authentication and security (Needham).

Work on theorem proving, e.g. Isabelle (L.C. Paulson).

Computing Service adopted Granta Strategy to promote distributed computing: based on Granta Backbone Network, optical fibre cabling across University and colleges, and combining mainframe with other computers and personal machines.

1989 Full 3-year Computer Science Tripos began.

IBM 3081D upgraded to 3084Q; Automatic Cartridge Store.

1990 Teaching and research staff 27. Computing Service established staff, including engineers, 44. Total Laboratory staff 127. About 30 postdoctoral research fellows and assistants.
92 PhD students, 33 Diploma, 170 Tripos, 19 MPhil.
About 6500 active users.

1990 onwards
Increasing emphasis on multimedia computing.

1992 Granta Backbone Network completed (a political achievement for Hartley as well as technical achievement by him and his team).

1993 Cambridge Honorary Degree for Wilkes.

Autostereo display (Wiseman and S.R. Lang).

1994 Hartley succeeded as Director of Computing Service by M.D. Sayers.

Professor D. J. Wheeler retired.

1995 N.E. Wiseman, an early worker in the laboratory, later Chief Engineer and subsequently on the teaching staff, died in service.

IBM 3084, last general-purpose mainframe in the Laboratory, decommissioned.

1995 A.J.R.G. Milner appointed to Laboratory's first established chair; succeeded Needham as Head of Department in 1996.

1996 Hitachi SR2201 parallel machine housed in Laboratory.

1997 Research Assessment Exercise Grade 5* (top, as in previous three).

Link with Cambridge Microsoft Research Laboratory.

William H. Gates III Foundation benefaction for new building.

BCS Award 1997 for 'Iris Recognition' to J.G. Daugman.

1998 Main lines of research during 90s:
theory and formal methods, theorem proving;
compilers, interpreters, and program analysis;
distributed systems and communications, multimedia;
database structure and systems;
security, authentication and privacy;
graphics, animation and interaction (including 3D display);
natural language and information processing;
vision and image processing.

Many and varied collaborative connections with industry, other universities, etc., in UK and abroad; continued input to local start-up companies.

Teaching and research staff 29. Computing Service established staff, including engineers, 54; full staff 93. Total Laboratory staff 134. About 35 postdoctoral research fellows and assistants.
105 PhD students, 42 Diploma, 259 Tripos, 22 MPhil.
About 24000 registered users: 'everyone in the University and then some' (Unix service 5000 users, mail service 22000, workstation filestore 9000).

Past members of the Laboratory are now be found in prominent positions all over the US and the UK, in companies and universities. Honours for members of the Laboratory have included two Turing Awards and fellowships of the Royal Society, the Royal Academy of Engineering and the British Academy.

This is version 5.3 of the History, reflecting its state on 1 April 1999.

Acknowledgements

We would like to thank the many people who have helped, in many different ways, with EDSAC 99:

- All our speakers;
- the Overseers of major activities –
  Chris Hadley, Simon Moore, Graham Titmus and Richard Stibbs;
- the EDSAC 99 Secretary – Beth Want; and
- Peter Crofts, Celia Denton, Rebecca Isaacs, Piete Brooks, Nick Batterham, Clare Seamark, David Greaves, Margaret Levitt, Sue Edmonds, Bruce Godfrey, Angie Pople, Steve Pulman, Sylvia Knight, David Abensour, Larry Paulson, Giampaolo Bella, Katherine Easthaughffe, Alexis Hombrecher, Andrew McNeil, John Daugman, Richard Watts, Paul Cunningham, Steev Wilcox, George Taylor, Myra van Inwegen, Paul Menage, Dickon Reed, Tim Granger, Austin Donelly, Richard Mortier, Neil Strafford, Philippa Gardner, Trevor Boyd, Alan Mitchell, Chris Town, Colin Watson, Matthew Wakeling, Neil Dodgson, Andy Penrose, Jonathan Pfautz, Michael Blain, Marco Gillies, Tony Polichroniadis, James Gain, Brett Saunders, Herbert Bos, Stephen Childs, Brian Cowe, James Hall, Joe Hurd, Abida Khattak, Jon-Hyeon Lee, Fabien Petitcolas, Umar Saif, Alistair Turnbull, Simon Bates, Naeem Khan and Staff of the Photography and Illustration Service.

We are also grateful to Joyce Wheeler, and to the University's Development Office and the Whipple Museum.

EDSAC 99 was recorded on video by Cambridge University Moving Image Studio. The video for the Engineers' presentation was made by Blair Hartley Limited, Chesham.

For information about copies of this booklet and the video proceedings of the EDSAC 99 meeting, please contact the Computer Laboratory.

Postscript – the Bun Shop

The development of the EDSAC was punctuated by a series of landmarks. … These landmarks we celebrated by a journey to the local pub – known as the Bun Shop – where I would treat my colleagues to a pint of beer.
[M.V. Wilkes, Memoirs of a Computer Pioneer, MIT Press, 1985, p142.]