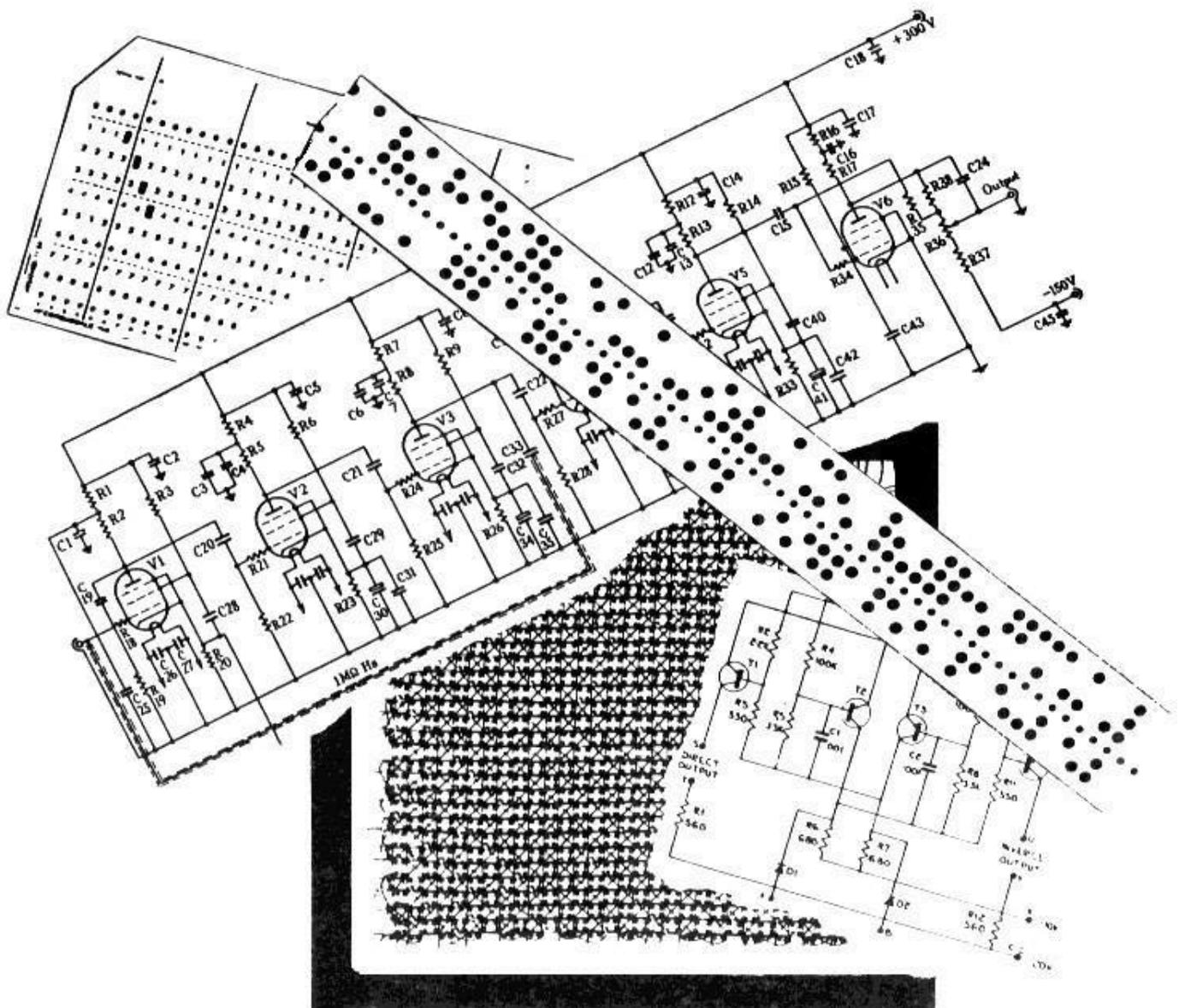


Computer

RESURRECTION

The Bulletin of the Computer Conservation Society



Computer Conservation Society

Aims and objectives

The Computer Conservation Society (CCS) is a co-operative venture between the British Computer Society (BCS), the Science Museum of London and the Museum of Science and Industry in Manchester.

The CCS was constituted in September 1989 as a Specialist Group of the British Computer Society. It is thus covered by the Royal Charter and charitable status of the BCS.

The aims of the CCS are to

- ◇ Promote the conservation of historic computers and to identify existing computers which may need to be archived in the future,
- ◇ Develop awareness of the importance of historic computers,
- ◇ Encourage research on historic computers and their impact on society.

Membership is open to anyone interested in computer conservation and the history of computing.

The CCS is funded and supported by voluntary subscriptions from members, a grant from the BCS, fees from corporate membership, donations, and by the free use of the facilities of both museums. Some charges may be made for publications and attendance at seminars and conferences.

There are a number of active Working Parties on specific computer restorations and early computer technologies and software. Younger people are especially encouraged to take part in order to achieve skills transfer.

Resurrection

The Bulletin of the Computer Conservation Society

ISSN 0958-7403

Number 45
Winter 2008/9
Contents

Doron Swade MBE <i>David Hartley and Dik Leatherdale</i>	2
Editor's Remarks <i>Dik Leatherdale</i>	3
News Round-Up	4
Society Activity	9
Pioneer Profiles – Tony Brooker <i>Brian Napper</i>	11
Some CAFS Applications <i>Hamish Carmichael & Martin Wright</i>	14
Where did it all go? <i>Simon Lavington</i>	24
Obituary: F.J. Murray Laver <i>Reay Atkinson</i>	30
You Can Become a Computer Programmer! <i>Dik Leatherdale</i>	31
Forthcoming Events	32

Doron Swade MBE

David Hartley and Dik Leatherdale

Our first happy task in this edition of *Resurrection* is to congratulate Doron Swade on behalf of the CCS on the award of an MBE for “Services to the History of Computing”.

Doron is probably best known as the world’s leading authority on the work of Charles Babbage and as the leader of the team that



Doron Swade & Difference Engine No. 2

constructed Babbage’s Difference Engine No. 2.

As Curator of Computing at the Science Museum for 14 years, he has raised the profile of computer history by his numerous publications and public appearances as well as making several important acquisitions for the Museum.

But his most important contribution, from the perspective of the Computer Conservation Society, is that the Society was his idea. Doron it was who approached the BCS with the notion that a society might be formed and, as a result, nearly 20 years on, we are now a thriving and authoritative group supporting the field to which Doron has made such a hugely significant contribution.

Resurrection has published several articles from Doron over the years including an article in the very first edition. Those of us who were privileged to attend his recent lecture on his home ground at the Science Museum, can testify to the immense contribution he continues to make and to his remarkable talent for inspiring enthusiasm for the subject.

The MBE citation mentions all these activities and more. Doron modestly states “It is flattering and humbling to be recognised in this way, and the award is terrific for the field of history of computing”. Indeed it is, but it is mainly a well-deserved honour for a distinguished individual.

The Computer Conservation Society salutes Doron Swade MBE.

Editor's Remarks

Dik Leatherdale

So, welcome to the first edition of *Resurrection* in 2009, a year which sees the 60th anniversary of EDSAC, with the Apple Macintosh, the ICL 2900 and the IBM 360 celebrating 25, 35 and 45 years respectively. It is commonplace to observe that the pace of change in computing continues to accelerate, but EDSAC and all its contemporaries were consigned to the history books comparatively quickly. Perhaps the commonplace view is not altogether correct.

In a November edition of *University Challenge*, Jeremy Paxman asked "In 1951, Christopher Strachey wrote the earliest successful game-playing program to demonstrate artificial intelligence. What game did it play?". Sadly, the correct answer was not forthcoming. But had the contestants read *Resurrection 43*, they would certainly have known the answer.

Later the same month, contestants were asked to identify the father of Ada Lovelace.

Clearly, computer history has become one of *University Challenge's* recognised subject areas. But which of our readers is their subject matter expert?

Readers will recall Michael Knight's article on Beacon in *Resurrection 43*. Michael's article was republished by the *VIP Club* as their 'article of the month'. The *VIP Club* is the organisation for former staff of Unisys and Lockheed Martin. Its website vipclubmn.org is an excellent resource for anybody interested in the history of Univac. Our congratulations to Michael.

The profile in this edition features Tony Brooker who, as you will read, pioneered the development of high level programming languages in the early 1950s and much else besides. If readers detect a moment of self-indulgence on my part in the selection of the subject, they are entirely correct.

North West Group contact details

Chairman Tom Hinchliffe: Tel: 01663 765040.

Email: tom.h@dsl.pipex.com

Secretary William Gunn Tel: 01663 764997.

Email: william.gunn@ntlworld.com

News Round-Up

The last remaining PDP 11-based air traffic control system from West Drayton near Heathrow has been acquired by the National Museum of Computing and is now working again at Bletchley Park.

The system comprises two of these hugely impressive display screens, each driven by a PDP 11/34.



The main application runs on two PDP 11/84s, which, together with various pieces of ancillary equipment, are seen below.



Milton Keynes is not yet blessed with a major international airport, so there can be no connection to real, airborne events. A recording made on 14th March 2003 simulates the outside world and is processed by the PDP 11s, just as it would have been on that date.

We hope to be able to tell the story of the re-birth of the system in more detail in a forthcoming edition of *Resurrection*.

101010101



An ICL 2966 has been installed at the National Museum of Computing and, with excellent and generous support from Fujitsu Services, is coming back to life. The Order Code Processor (OCP) has been powered up and has been confirmed to be working. One of the engineers working on the 2966 is the engineer who decommissioned it 15 years ago.

101010101

Thrilling news from Bletchley Park that English Heritage has made a grant of £330,000 which will enable the remaining repairs to the roof of the mansion to be carried out. The repairs will safeguard the historic interiors from damage and puts the project on a sound footing. The work has already started and will be complete by spring 2009.

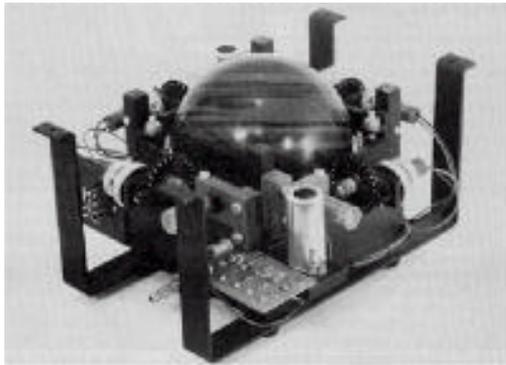
In a separate announcement, a further English Heritage grant of £100,000 for each of the next three years has been given to allow the restoration of the various huts in which much of the wartime codebreaking took place. This grant, however, is subject to the acquisition of matching funding from other bodies.

101010101

The 9th of December 2008 marked the 40th anniversary of the first public demonstration of the computer mouse by its inventor Doug Englebart. <http://news.bbc.co.uk/1/hi/technology/7772376.stm> is a film of the event.



At much the same time Logitech claimed to have sold their billionth mouse.



Sadly, nobody seems to have remembered that the trackball preceded the mouse by almost 20 years, invented in Canada by Tom Cranston and Fred Longstaff of Ferranti Packard. Longstaff went on to be chief architect of the FP6000 computer which many CCS members will remember as the ICT 1900.

101010101

In the Victorian splendour of Manchester University's Whitworth Hall, Mr Tom Bloxham, the first Chancellor elected since the unification in 2004 of the Victoria University and UMIST, was installed with due ceremony. This was followed by the conferring of Honorary Doctor of Science degrees on Sir Tim Berners-Lee, of World Wide Web fame, and on Sir Bernard Lovell, Radio Astronomer. Two other men were also honoured. CCS member Conway Berners-Lee was present to see the award to his son, and reference was made to the continuity from the "Baby" computer, through the Mark I on which Sir Tim's parents worked, to the present day and to Sir Tim's work on the Semantic Web. Sir Bernard Lovell, at age 95, responded on behalf of the honorary graduands, and made an excellent speech, without notes.

101010101

In November a small team of CCS members visited the Museum Collections Centre in Birmingham to attempt to locate and identify the HEC1 computer, the WITCH computer, a Booth magnetic drum, and Booth's electromechanical storage unit.

In 2002, a previous visit by Roger Johnson, Dan Hayton and Chris Burton had seen and examined all but the WITCH in a dirty and untidy condition at the Charlotte Street store, a derelict former print works. The new Collections Centre is superb, with five-high pallet racking in clean

and light modern buildings. A staff of about six people looks after the thousands of artefacts. Most of the items are connected with Birmingham and Midlands art and industry in some way. The material is merely stored — there is no attempt to display or interpret it. Some objects at ground level have description cards. The Centre is open to the public once a month. Because of the sorry state of preservation of items when last seen at the old Charlotte Street store, we were anxious to make sure that particularly the Booth drum and storage unit had survived and were safely looked after.

Kevin Murrell had kindly setup this visit. He and two other of our members, Tony Fraser and Johan Iverson, were particularly interested in the WITCH computer. Built by E. Cooke-Yarborough, WITCH had started life at Harwell and made extensive use of Dekatrons (cold cathode decade counting tubes) for storage. It was donated to Wolverhampton College of Technology and gained its name as an acronym for Wolverhampton Instrument for Teaching Computation from Harwell. After its useful life as an educational object was over, it was donated to Birmingham Science Museum. For more information see <http://www.computerconservationsociety.org/witch.htm>.

Dr Raymond ‘Dickie’ Bird also attended, with Roger Johnson and your correspondent — Chris Burton. We were keen to view the HEC1 again, particularly because Dickie had built it at BTM in about 1951 from drawings of the original APE(R)C designed by Andrew Booth at Birkbeck College. See *Resurrection* 22.

The Head of Collections at Birmingham, Phil Watson, had arranged to bring the HEC down to floor level for us to examine. The four racks were packed face to face on a stillage, which Phil agreed to have unpacked and arranged in a row for us to see. Of course, Dickie Bird was delighted to see his handiwork again almost 60 years after its creation.



HEC 1

We then searched for the separate magnetic drum and electromechanical store unit. We were assured that these were somewhere in the building, probably in one of the hundreds of boxes in an adjacent area of racking. We drew a blank, and will need to do further investigation to find them. The likely problem is that they have been mislabelled and catalogued incorrectly. We did notice several objects with descriptive labels which are manifestly wrong, presumably due to understandable lack of specialist expertise at the Collections Centre. For example, one object labelled as “a control panel for an ICL 1400 computer” is actually a factory test unit for Ferranti Orion logic packages.

Meanwhile the other half of the party had located the WITCH, consisting of several sub-assemblies packed in behind a stone well-head from Venice, an Orion control desk, a variable-speed motor control display and a neo-Gothic chancel screen in carved oak. It was very difficult to get at the units, and extremely hard to get a reasonable photograph. It seemed that all the parts were present except for the Input/Output equipment, which may well be elsewhere in the Centre. Johan persevered in trying to get as many photographs as he could. In passing, we noted that the drawers of the Orion control desk contained a large number of paper tapes of test programs, as well as magnetic tapes. In the same area was a very early, possibly prototype, Elliott film handler, and a sad-looking Samastronic printer.



The relay logic of WITCH

Our party then adjourned to the documentation room where the catalogues and records are kept. Several documents pertaining to WITCH were photographed, together with a few relating to the HEC.

The visit was heartening in that the objects are now cared for in a systematic and tidy environment, albeit that cataloguing and descriptions may need some more attention, particularly in our field of interest. It is interesting that some little-known artefacts are there to see.

The Birmingham Museum Collections Centre has free guided tours on the last Friday of the month, with advance booking, and occasional Open Days. The next one is on Monday 4th May 2009.

Society Activity

Bombe Rebuild Project

John Harper

The appeal that we made in *Resurrection 44* has already resulted in a very generous £500 donation. This has allowed work to continue on the extra 45 drums of the three additional types. All covers are now painted in the three colours: cobalt, silver and black. One of each of these is the variant used on our Checking Machine and they have been completed and tested. It is therefore possible for us to check out a menu that contains any three of eight wheel combinations. Obviously, this is working backwards because it means that we know that the setting had already been found and all we are doing is rechecking. However, when we have all the drums complete, we will be able to rerun original menus and, just maybe, break messages not previously broken during WWII.

We would like to repeat that we are still seeking sponsors who could be identified with the completion of this valuable enhancement. To complete this work we now need £1500.

In addition to re-running a variety of WWII decrypts, we have been considering adding other original features. The three extra drum types are part of this. However, in addition, we are now in the process of making a fourth Diagonal Board. This is similar to the three originals but has extra jacks that provide SSS (Single Self Stecker) and CSKO (Consecutive Stecker Knockout) features. When the assembly is completed and tested, I will describe them in more detail and will also show how they were used. The Diagonal Board Frame can be fitted instead of one of the originals or, alternatively, on temporary supports as a fourth unit.

Over the years we have been involved in many filming exercises but the outcome has not been very satisfactory. Usually, the technical commentary also leaves much to be desired. But we were very pleased with the outcome of filming that took place this summer. For perhaps the first time, the descriptions of the Enigma code breaking process at Bletchley Park and how much the Bombe figured in it are very much to our liking. It is a National Geographic production entitled; 'CODES: Murder, War and Treason'. It has been shown from time to time on the National Geographic Channel but unfortunately only on satellite or cable TV. Perhaps it will be shown on other channels in the fullness of time. We recommend that you look out for it.

Our website is still at www.jharper.demon.co.uk/bombe1.htm

Pegasus Working Party

Len Hewitt and Peter Holland

The demonstrations of the running of Pegasus have continued at the usual fortnightly intervals without any significant problems. We are now able to duplicate the punched paper tape output by displaying it in suitable large characters on a PC screen at the same time as it is punched out on tape. This makes it easier for visitors to see the output of programs without waiting for it to be printed and the facility is also useful in the case of any trouble with the usual electromechanical output devices. We have now contacted a Creed engineer who has visited us briefly. He will return in 2009 when we hope he will be able to help us with our tape preparation equipment and other Creed devices.

Our first “In Steam” day of the New Year will be Wednesday, 14th January and then, at fortnightly intervals on every other Wednesday, the computer will be switched on from 11:00 to 15:00.

Contact Len Hewitt at leonard.hewitt@ntlworld.com.

CCS Website Information

The Society has its own website, which is located at www.computerconservationsociety.org. It contains news items, details of forthcoming events, and also electronic copies of all past issues of *Resurrection*, in both HTML and PDF formats, which can be downloaded for printing. We also have an FTP site at [ftp.cs.man.ac.uk/pub/CCS-Archive](ftp://ftp.cs.man.ac.uk/pub/CCS-Archive), where there is other material for downloading including simulators for historic machines. Please note that this latter URL is case-sensitive.

Contact details

Readers wishing to contact the Editor may do so by email to dik@leatherdale.net, or by post to 124 Stanley Road, Teddington, TW11 8TX. Queries about all other CCS matters should be addressed to the Secretary, Kevin Murrell, at kevin@ps8.co.uk, or by post to 25 Comet Close, Ash Vale, Aldershot, Hants GU12 5SG.

Pioneer Profiles – Tony Brooker

Brian Napper

Tony Brooker graduated in Mathematics from Imperial College in 1945. He stayed at Imperial for four years, during which he built a relay-based multiplier for the first Imperial College Computing Engine. In 1949, he moved to the Mathematical Laboratory at Cambridge, looking after the Differential Analyser and learning programming under David Wheeler, with whom he implemented an interpretive floating-point scheme.



In 1951, Tony moved to Manchester University to take over as head of software development from Alan Turing, on the recently installed Ferranti Mark I.

From 1948 to 1963, Manchester University under Williams, and then Kilburn was continuously at the forefront of computer hardware development worldwide, with the Ferranti group at Manchester turning three of their designs into commercial machines – Mark I (1951), Mercury (1957) and Atlas (1962). Tony Brooker made sure that programming language development kept pace.

In 1951 the Mark I was the only machine with a fast random-access secondary store, the drum, the precursor of the ubiquitous disc. On arrival, Tony found a reasonable load time and run time system in place, Scheme A, but a method for inputting actual program that was a triumph of ease of implementation over ease of use! Turing had devised a system for writing programs directly in binary, using a 32-base numbering system, with digits 0 to 31 represented by the characters, in order, /E@A:SIU%DRJNFCKTZLWHYPQOBG"MXV£. These were essentially the symbols corresponding to the five-bit patterns on the paper tape used for input and output. All that was needed to read a line of program was to read the four symbols specifying the instruction into the next 20-bit word!

Tony immediately improved the load/run time system, with Scheme B. But instead of designing a better assembly language, Tony went on to design the Mark I Autocode, (1954), a pioneering high-level language.

In the Autocode, a line of program corresponded to a basic operation required for a computation, e.g. $v3 = v6 + v8$. The user was provided with a set of floating-point variables, $v1$ to around $v5000$, a set of

integers `n1` to `n18`, a method of indexing variables, e.g. `vn5`, labels, and some built-in functions. The Mark I had only fixed-point arithmetic, so arithmetic involving variables was implemented by subroutines. Variables were held on drum, but the user was not made aware of this.

Not surprisingly, the Mark I Autocode made programming accessible to a much wider audience, making it far easier to write straightforward computation programs, although they ran much more slowly. It inspired Pegasus Autocode, which introduced programming to large numbers nationwide.

When the Mercury arrived in 1958, Tony provided Mercury Autocode. Efficiency was now an issue. Mercury provided floating-point by hardware, but the two-level store had to be made visible to the user. A large program had to be divided into subsections of limited size, called Chapters, with their own local store, with control passing between them explicitly. A large auxiliary store of variables was provided, with transfers between local and global store programmed explicitly. The language was now more sophisticated, e.g. with loops and declarations, but still below that of the first Fortran (first available in 1956). The language was later upgraded with EMA, Extended Mercury Autocode.

For the next Manchester machine, Atlas (1962), Tony designed Atlas Autocode. Fortran II (1958) was now becoming a worldwide standard, and Fortran IV and Algol 60 were on the way. Atlas Autocode followed the Algol model. The language level was a bit below Algol's, but with better efficiency of compilation and execution.

With its massive increase in speed and size over Mercury, Atlas required a revolution in hardware techniques to create it, and in operating systems to cope with its massive throughput. The contemporary STRETCH and LARC in the US needed similar leaps. Atlas Autocode was a major advance compared to EMA. But Brooker's unique contribution to software development was the Compiler Compiler (CC), an innovative language and implementation with few equals worldwide at the time. Derrick Morris was, by now, a key assistant.

Given that at least five languages would be required for Atlas it was decided to speed and harmonise the process of writing compilers by designing a special-purpose language. CC provided a simple basic language for handling integers and symbol strings. On top, it provided sophisticated facilities for the definition, recognition and processing of syntactic elements, based on BNF, and it introduced, in effect, a new data type, the "phrase variable", by which syntactic elements could be processed efficiently inside routines, and passed as parameters.

To give a minimal example, a statement in a language could be

```
CYCLE [NAME] = [EXPR] [STEP] TO [EXPR]
```

Where [STEP] is defined, just for use in this statement, as

```
PHRASE [STEP] = UP, DOWN, BY [OPERAND] UP, BY [OPERAND] DOWN
```

In the CC routine, translating a source statement of this form might be the pair of statements

```
JUMP 14 UNLESS [STEP/1] BY [OPERAND/2] DOWN
```

```
COMPILE [NAME/1] = [NAME/1] - [OPERAND/2]
```

A phrase variable is written in square brackets, giving the syntactic element type and an integer to distinguish between different variables of the same type within a routine.

The language processing machinery was available within CC itself, so a similar example could refer to a compiler-writer adding his own informal cycle statement to the CC language!

The internal structure of CC was very flexible, and it was implemented as a sophisticated bootstrap, only requiring the built-in octal loader of Atlas to start. It could be completed in a single continuous run. Key stages included : simple assembler in octal – basic CC machinery in assembler – basic CC machinery redefined and refined in basic CC – full CC in basic-to-full CC. Routines and phrase definitions could be defined or redefined at any time with instant effect (with care!). Even the compiler-writer's program was treated as an extension to CC.

CC was successful in enabling the official compilers for Atlas to be written more quickly and securely. But these compilers ran slower than if written in assembly language. CC spawned a number of similar systems, most notably C's YACC (Yet Another Compiler Compiler!). But in general, they downsized the language facilities to avoid the inefficiency caused by the unrequired power and generality of CC, and phrase variables were not implemented.

Having seen through the establishment of the software on Atlas and helped the creation of the first UK undergraduate Computer Science department, Tony left Manchester in 1967 to take a chair at Essex and run the new "Computing Science" department there. Your editor is one of its first graduates.

Having awakened an unexpected interest in and talent for university governance, Tony rose to be a pro-vice chancellor before retiring in 1988. He now lives quietly nearby.

Some CAFS Applications

Hamish Carmichael & Martin Wright

CAFS could, as the marketing of the time put it, “find a noodle in a hat rack”. Not just a question of the innovative hardware, but also the surrounding software and its application.

This article is, in effect, a continuation of the presentation on the ICL CAFS System given to the Society by Vic Maller and others on 28th January 1993, in which the functionality of CAFS was fully described, so the introduction is restricted to the briefest overview.

CAFS was – and indeed still is – a *search engine* implemented in hardware. The term is now regularly abused by Google and the like, who go to enormous lengths to avoid any actual searching. Their indexes and concordances are immensely clever and wonderfully efficient, but in the end, they take you straight to the relevant pages – because they know where they are – they don’t have to go looking for them. And the process uses immense amounts of raw computing power.

But thirty years ago, when a mainframe, struggling to support a family of terminals, had less power than your average modern wristwatch, such access-enabling strategies were practically, and certainly economically, impossible. For ad hoc enquiries we often had to search our files. And searching was a slow, laborious and anti-social thing to do. You had to heave the file, block by block, from disc into mainframe store, unpack the records, and then plough through them by software, probably discarding 95% of them. It was enormously wasteful, and the path-lengths were horrendous.

CAFS let you say which records you wanted, on the basis of their content, and then it would get them for you. All the searching happened way down in the disc controller, and the application in the mainframe would only see the required hit records. Magic!

I first encountered CAFS in the mid 1970s. By that time I with many others had been engaged for over 10 years in computerising ICL’s basic administrative systems. In this context, I was made aware of CAFS and at once became an ardent enthusiast. The developers had reached the stage where they wanted to try their machine out on a real file and applied to us for help; they could hardly go out to a customer, so it made sense to look internally for a convenient file of searchable data. It was decided that the Personnel file, at 45 megabytes, was a suitable size so, after some

judicious scrambling of personal details, a copy was taken up to Stevenage for the team to play with.

Perhaps we should explain that, at this stage, the CAFS project was still very experimental and was not highly thought of in Putney. Corporate Marketing was then in thrall to a fat Australian slob whose sole policy idea was to watch what IBM were doing. (But the trouble with a policy of following IBM is that ipso facto you're always behind.) He argued that IBM didn't have a CAFS, so why should ICL have one? He didn't seem able to understand that CAFS would make an ICL system more efficient and that this would give us an advantage. Our argument was that IBM would never implement a CAFS because it would make their systems more intelligent and efficient, and that would interfere with their policy of covering the world five feet deep in dumb blue boxes. So, for top ICL management, CAFS was suspect, and indeed when Chris Wilson was Managing Director, he thought he had killed it off altogether. But Gordon Scarrott was a wily bird who kept it going out of sight and under the counter.

The ICL/IBM Comparison

Although it refers to CAFS-ISP and is thus out of chronological sequence, it makes sense to include the description of an interesting experiment at this point. We got a friendly consultancy firm to mount the same file on three separate systems – an ICL 2966, an IBM 3081 and an IBM 3033 – and run the same enquiry on all three, from three terminals mounted side by side, as a race, with these dramatic results:

	2966	3033	3081
Raw processor power (MIPS)	1.1	8	10
Average response time (seconds)	48	219	183
Mill-usage per enquiry (ms)	0.8	16	33
Cost per enquiry	£0.60	unknown	£160

Sadly those of us were CAFS enthusiasts never managed to get that startling contrast in performance properly understood by Corporate Marketing or by the Directors of the company.

While the experiments with the Personnel file were going on, one thing that was particularly striking was the attention the CAFS research team paid to nuances of language in the way users expressed their enquiries. For example, imagine Morse saying to his sidekick : “Lewis, get me a list of all the blue and green Fiestas in Oxford”. Well, we all know that he didn't mean Fiestas that were painted both blue *and* green at

the same time; he meant Fiestas where the colour was either blue *or* green. So how could software detect in the enquiry the latent *or* behind the expressed *and* ? Consideration of such questions was a fundamental part of the CAFS development environment.

The ultimate CAFS enquiry software had linguistic subtleties that were way beyond any contemporary rival. From watching users' preferences it offered stem-matching as the norm and 'Exact' matching as an option. In specifying a range of values one could distinguish between an inclusive 'From – To' and an exclusive 'Between – And'. There was the built-in Quorum facility: one could say 'Get me records which satisfy any three out of these five conditions'. Indeed one could attach numeric weights to individual conditions and specify the total score that a record had to achieve in order to be counted as a hit. And there were lots of subtle assumptions to make it as easy as possible to understand and use, with the minimum of typing for the user. By contrast, one has only to think of SQL, which, like many software products in the IBM world, was never designed for use by human beings

Another most interesting fact was that CAFS was designed to search equally effectively on formatted data and text. At the time there were data systems and there were document storage and management systems, but none providing facilities for both worlds. For proper documents IBM had STAIRS; everybody else had STATUS. The records in our data files might contain textual elements, but that text was dumb and useless, only carried around with the data until it could be printed out. Now it could be used in enquiries!

The time quite soon came where Personnel management demanded CAFS access to the live Personnel file, not just to an experimental copy. And, as soon as they got it, the world changed. The Personnel Director confessed that he was used, in Board meetings, to being the bad guy – the one who delayed decisions because he would have to go back to the files, do an analysis, and report back in six weeks' time. Now, when a question arose (sadly it was usually about whether this or that redundancy programme was more viable), he could nip out to a terminal in the anteroom, ask the questions, and reappear with the answers before all the coffee cups had been refilled. The status of the Personnel function rose like a rocket.

So in 1979 it was decided to release a CAFS product to market. It came in the form of the CAFS 800, a wardrobe-sized first generation product, packaged with a 2946 processor running the 1900 DME operating system. The content addressable store was held on up to 14

60MB exchangeable discs; (they seemed large at the time). Because the CAFS search mechanism itself was much faster than the disc transfer rate available, some of the disc units were modified so that ten tracks could be read simultaneously, thus bringing the transfer rate, and hence the searching rate, up to a respectable three megabytes per second.

The searching software was prepared for public release by the Bracknell software factory. Unfortunately, in productising the enquiry package transferred from the Stevenage labs they lost some of the finer linguistic subtleties, but the issued General Enquiry Program was still a superb package, flexible, varied and easy to use. Discerning customers began to take advantage of it.

CAFS 800

The North Yorkshire Police was our introduction to the Police world with which we became increasingly familiar.

Its first use of CAFS occurred almost immediately after the system had been delivered, and took place during the visit of Pope John Paul to York racecourse. This pilot application dealt with lost and found property, and reuniting lost children with their parents. (With striking commonsense the children usually said that it was their parents who were lost.)

Its success led immediately to the development of a comprehensive Incident Logging System which was introduced across the Force in July 1983, recording all incidents, including Crimes, Accidents, Beat Complaints, Missing Persons, and Lost and Found Property. With this CAFS jumped straight into the main line of police activity. Crucial factors were the force-wide availability of the system; the information wasn't restricted to those pen-pushers at Force HQ. And because users at all levels got advantage from using the system, data entry to it was quickly recognised as being important rather than a chore to be left till later. The brochure, which a proud Force helped us to produce, also stressed the speed of implementation of the system and that the need for training in its use was much less than they had expected

The Civil Aviation Authority used CAFS 800 for its Occurrence Reporting System, which not only held details of official reports but also the multifarious informal reports that could be submitted by anyone involved at any level in the operation of aircraft in British airspace and at British airports. The previous conventional database application was slow in handling ad hoc enquiries, couldn't enquire on the text in incident reports, and didn't allow browsing in the search for trends.

Putting it on CAFS dramatically improved accessibility and response, and radically reduced the demand for ad hoc printed reports – which had previously held back the development of other main applications. This reduction in demand for ad hoc reports was reported from virtually all the early CAFS implementations.

The private telephone company operating in Hull was also quick to adopt CAFS 800, perhaps partly from a desire to stress its difference from the Post Office. CAFS enquiries became an essential part of many of its day-to-day administrative systems. As an example, its advantage in the in-house analysis of network maintenance effort was stressed.

Hull also used CAFS 800 for Directory Enquiries. In this it followed the Post Office, which had conducted an extended experimental service in the earlier stage of the CAFS story, before ICL had taken the decision to launch the product. That experiment was brought to an end because the Post Office could not afford to base a vital part of its service on technology which was not yet fully approved by its own makers. The story is told that there were copious tears in the exchange in Bedford when the experimental service was closed down and the operators realised that they were going to have to revert to answering enquiries by referring to the paper phone directories.

Computel was a Bracknell company running a conventional bureau service for a variety of clients. It adopted CAFS 800 with the intention of offering a dial-in enquiry service to users on their own data files. For various reasons it was never an out-and-out commercial success. However, Computel's involvement was important for examining the possibility of integrating CAFS capabilities with other technologies – eg IDMS, Viewdata, and even Word Processing

Individuals from Hull Telephones and Computel became important members of the CAFS consultancy brotherhood.

Within ICL itself, the success of the Personnel application led the Corporate Systems organisation to develop a variety of other enquiry services, and the value of the CAFS idea was progressively much more widely understood

In total there were some 20 CAFS 800 installations, which was a pretty good record for such a non-mainline product. Some of them kept going for as long as eight years.

The transition to CAFS on VME systems

But there was one strategic snag with CAFS 800. It worked in the old 1900 environment, with six-bit characters, where text was case-blind, and everyone could see that the future lay with bytes, the 2900 architecture, and the VME operating system. Of course the 1900 environment had been ideal for the development of such a revolutionary product; one could get at the hardware and tell it exactly what to do. VME, by contrast, has been described as a system “designed to prevent the programmer taking advantage of the hardware”. More importantly, in its early years VME certainly didn’t provide a stable enough environment to support a project combining experimental hardware and software.

Another problem was that the CAFS 800 software was read-only. Thus the data which CAFS was to search had to be maintained by some other software and then copied into searchable form at suitable intervals. This was acceptable where the data was relatively static or at least slow moving, but we and our customers were increasingly aware that the biggest benefit would come when CAFS searching capability was applied to highly volatile data.

The company decided that, yes, there should be a CAFS in the 2900 world, and an excellent team at West Gorton designed the hardware. In typical second generation fashion, the size went from wardrobe to shoebox, and the resulting module could be accommodated within the frame of a standard disc controller. Out went the complication of reading from many tracks simultaneously, but disc transfer rates had gone up, and the resulting search rate of 1.2 megabytes per second was deemed acceptable.

Everybody wanted to minimise any disturbance to VME itself, so there was to be no CAFS-specific low-level software. Instead CAFS capability was neatly hidden inside the existing index sequential access mechanism. The initiation of a CAFS search simply appeared to that mechanism as the outward transfer of a record. As a minor consequence the quorum facility, though still present in the hardware, was not fully supported at the software level.

Within Marketing, some argued for retaining the name ‘CAFS’ on the ground that this was already well understood in the user world. Others said the product ought to have a name which more clearly described its function, and proposed to call it the ICL Information Search Processor. By a masterly compromise it was finally decided to adopt the hybrid name ‘CAFS-ISP’.

The sparks started to fly when it came to the design of the user-level software to go with the new CAFS. Bracknell, where software came from, had gone all relational and, ignoring all the linguistic and psychological experience of Stevenage, had produced an SQL-like package which was touted as the company's official offering. So the new CAFS-ISP came out into the world with the relatively dismal Querymaster, which was soon supplemented by two programming tools – the Relational CAFS Interface and the Direct CAFS Interface – and by the even simpler CAFS Search Option extension to 2900 System Control Language.

Querymaster, RCI, DCI and CSO

The Queensland Police Service became the first overseas user of CAFS-ISP, and the first foreign police user. It was fortunate in having in a key position a man with extensive police experience who was also fully up to date with computer capabilities; he had instantly seen the possibilities of CAFS 800, and switched to CAFS-ISP at the first opportunity. As he said: "Police forces need to be able to search huge quantities of unstructured data at high speed. You never know when a particular piece of information could become a key factor in solving a crime". QPS achieved great success with Querymaster as soon as they got it.

In the UK, Austin Reed relied on a big inventory file, updated overnight with the day's sales, controlling production ordering, delivery van scheduling, etc. Making it accessible via a CAFS service had instant results: "Suddenly everyone wanted a terminal. Everyone could see the benefits of access at such speed to their own data. Because it is the user who asks the question, the answer he gets has far greater credibility than those which used to emerge from the DP department."

Berkshire County Council already made extensive use of Viewdata to make a lot of council information available to the public. This included the library catalogue, covering over 300,000 titles, but access to it by Viewdata alone was too cumbersome and slow. It could find the book if the user specified the exact title and the exact name of the author, but most users couldn't manage this degree of precision; therefore some sort of fuzzy matching capability was essential. Berkshire achieved this by exploiting the Relational CAFS Interface, which "provided the general public with a quick and easy way to find the right book, whether in Reading or among the shelves of any Library operating throughout the county."

At Northampton Borough Council it was observed that for typical enquiries on the Income and Expenditure Annual Transactions file, which held one and quarter million records, response time dropped from two to three hours to two to three minutes, with an enormous impact on the usability of the system from the end-user's point of view. End users became another development resource – defining and executing their own enquiries instead of asking the computer staff for ad hoc reports. Use quickly spread to Environmental Health, Accountancy and Finance, Legal and Commercial, Housing Rents, Vehicle Maintenance, Housing Repairs, Technical Services, and the Establishment Section.

At Sheffield Metropolitan District Council a similarly wide range of departments used the CAFS facilities. Having mounted the Works Costing system, the IT department was surprised by how quickly the Works Department grasped the concepts and became expert exploiters. One of their Stores Auditors produced the best comment on the effect of the new enquiry service: “This is the Yes to everything you used to say No to.”

Southern Water reported many similar positive experiences. However, probably its most important contribution was in collating information from surveys of usage among a wide range of ICL customers, and helping to compile two excellent reports published by the ICL Computer Users' Association.

The Irish Motor Insurance company had an awareness of fraud remarkably similar to what we encountered later in Australia: “We recently searched with CAFS for all the registration numbers of cars that we knew had been ‘written off’ in the last few years. We could then eliminate them from our master files and analyse the people still claiming to own those cars.” Using Querymaster and CAFS, over 200 programs were no longer needed. In addition, many runs were dramatically speeded up by using the CAFS Search Option, where an existing program is used unchanged with a CAFS search being invoked by the controlling SCL

At the Oxford University Computing Service, Dr Lou Burnard reported: “I am enthusiastic about the use of computers in the humanities because there is so much untapped potential there. There are all sorts of areas where we can assist research workers, giving them the means to race ahead by speeding up the rate at which they can follow up a line of investigation. Where research is speculative and results-driven, the speed of CAFS is important because of the changes it makes possible in the characteristic modes of enquiry. Hypotheses can be tested as they are formed, and searches can be determined by the results of previous

researches. Our experience suggests that CAFS has a qualitative effect on research as well as a quantitative one.”

As an example, at the Oxford University Press editors had spent eight years trying to establish the most authentic and accurate text of Shakespeare’s work, combining the First Folio of 1623 with all 18 substantive Quarto texts. Once all these had been put into machine-readable form, the total corpus formed a database that could be searched end-to-end in 17 seconds flat, using less than a second of mainframe processor time. At least a dozen compositors worked on the First Folio, and one particular man had a penchant for semi-colons and used more than twice as many as any of his colleagues. By successive searches and by noting such anomalies as this the editors were able to pin down many errors in the original typesetting process of four centuries earlier.

IRAS, the Infra-Red Astronomical Satellite, was launched in 1983 and spent 10 months scanning the heavens, firing a colossal amount of information back to astronomers on earth. It increased the number of extra-terrestrial objects known to man by 40%, with positive identification of 250,000 distinct radiation sources. It took the scientists at the main American analysis centre about 20 minutes of mainframe processor time – i.e. an elapsed time measured in hours — for each full search through the IRAS Point Source Catalogue, so each search had to be treated as a batch processing job.

By comparison, at Queen Mary College scientists could run similar jobs interactively, getting results in two elapsed minutes for a fraction of the mainframe load. Dr David Walker explained: “Because we could use CAFS we were able to adopt a radically different approach, and it has proved remarkably successful. Even people who are using the most expensive supercomputers are only just managing to keep up with us.”

The foregoing examples show that CAFS could be applied as an enhancement or accelerator in many conventional application areas. Querymaster provided some facilities for online enquiries. The Relational CAFS Interface and the Direct CAFS Interface offered two ways of invoking CAFS searches from within a new program. The CAFS Search Option meant that an existing program could be speeded up by invoking a CAFS search from within the SCL statements that ran the program.

Editor’s note: This is an edited transcript of part of a talk given by the authors at the Science Museum on 18 October 2007. Contact Hamish Carmichael at hamishc@globalnet.co.uk.

Where did it all go?

Simon Lavington

Simon Lavington looks at some innovations in computing originating at Manchester University in the period 1948-74 and asks the question “What happened to them?”

Professor F. C. Williams took up the chair of Electrical Engineering at the University of Manchester in December 1946. He inspired a period of digital computer design that was characterised by fruitful collaborations between academia and industry. Five Manchester research prototypes and their commercial derivatives were produced in the period 1948 to 1974. In this article we assess what imprints, if any, the ideas embodied in these pioneering computers and their software have left on today’s Information Technology.

1. Origins and contemporary context.

In 1946, the concept of high-speed, automatic, general-purpose digital computers was gradually gaining credence in the scientific community. However, no such machine had yet been built. In 1949 Nathaniel Rochester (of IBM) remarked in an IRE editorial that “the most difficult problem in the construction of large-scale digital computers continues to be the question of how to build a memory, and the few papers written do not reflect the greatness of the effort which is being exerted”. Some of this activity had originated within defence establishments on both sides of the Atlantic, towards the end of the war

It is fascinating to trace the movement of people from British wartime research establishments to the universities and companies where, in the late 1940s, the pioneering digital computer activity took seed. The wartime centres were places such as the Admiralty Signals Research Establishment (ASRE) at Whitley, the Telecommunications Research Establishment (TRE) at Malvern and the Government Code and Cipher School (afterwards GCHQ) at Bletchley Park in association with the Post Office Research Station at Dollis Hill. The enthusiasm and electronic know-how that flowed from these establishments after the end of hostilities bore fruit in civilian organisations such as NPL and the Universities of Cambridge and Manchester, as well as in defence-orientated locations. Besides Manchester, the civilian computer projects that owed much to war-time know-how included the Pilot ACE at NPL

and the EDSAC at Cambridge. The Pilot Ace first worked in 1950 whereas EDSAC was working in 1949. In Manchester's case, Williams and Kilburn (from TRE) had a direct hand in the hardware design and systems architecture of the first Manchester computer, as described below. The mathematicians Newman and Turing (from Bletchley Park) influenced the software and use of the first Manchester computer.

F C Williams' first patent for electrostatic memory based on cathode ray tubes, filed in December 1946, originated from his work at the Telecommunications Research Establishment (TRE) at Malvern. His assistant, Tom Kilburn, moved with him to Manchester University. It was Tom, working for his Ph.D., who took the lead in refining electrostatic storage until, by the autumn of 1947, the Manchester group had demonstrated that one CRT could successfully store 2048 digits for four hours. To verify the basic memory technology, Williams and Kilburn built what they termed a Small-Scale Experimental Machine (SSEM, later called the Baby), a general-purpose computer with 32 words of CRT storage that first ran a program on the morning of Monday 21st June 1948. Though requiring manual input/output by means of handswitches and a display, the SSEM is believed to have been the world's first stored-program digital computer.

By mid-1951, 11 stored-program computers were in operation world-wide. In chronological order within country, these included five in America (SWAC, Whirlwind, ERA1101, UNIVAC and SEAC), five in Britain (Cambridge EDSAC, Manchester Mark I, Pilot ACE, Ferranti Mark I and LEO) and one in Australia (CSIRAC). Three of these machines used the Williams/Kilburn type of electrostatic memory, six used mercury acoustic delay lines, one (the ERA 1101) used a magnetic drum and Whirlwind used a special MIT-designed variant of electrostatic storage. Only the electrostatic technology offered true random access – by which we mean that access-time is independent of the physical location of a bit – for the high-speed primary memory. Furthermore, Manchester was the only group to integrate this primary RAM with a fixed-head drum backing store, thus introducing dynamic, software-controlled, swapping and overlays. The Cambridge EDSAC, first operational in May 1949, is believed to have been the world's first fully-functional stored-program computer in which, for example, input/output took place under program control. It was not until October 1949 that the Manchester Mark I had this facility, along with programmed transfers between CRT primary memory and drum secondary memory.

In Britain, the computer industry got off to an uncertain start. In September 1951 Professor Douglas Hartree had famously remarked to

Vivian Bowden, the first Ferranti computer salesman: “We have a computer here in Cambridge; there is one in Manchester and one at the NPL. I suppose there ought to be one in Scotland, but that’s about all”. However, by 1970 a dozen separate UK companies, together with six government establishments and four universities, had produced about 79 distinct computer prototypes. By 1970, these had resulted in about 2,200 installed British-designed production machines. The first non-British computer, an IBM 650, arrived here in October 1956. By 1967 IBM accounted for about 29% of all UK installations. ICT was formed in 1959. Ferranti’s main computer interests were acquired by ICT in 1963 and by 1968 ICT had become ICL and had absorbed all of the major UK mainframe computer companies.

2. The five historic Manchester computer projects.

Skipping backwards in time, Tom Kilburn gradually assumed leadership of the computer design group at Manchester University when Freddie Williams’ inventiveness turned to other areas of electrical engineering. Table 1 lists the five distinct projects to emerge from Manchester in the period up to 1974. Of course, other computer designs such as Dataflow followed later and indeed, Manchester remains in the forefront of computer science today. It is convenient to treat the five projects in Table 1 as a distinct set because each had close links with the market place at a time when there was such a thing as an independent British computer industry.

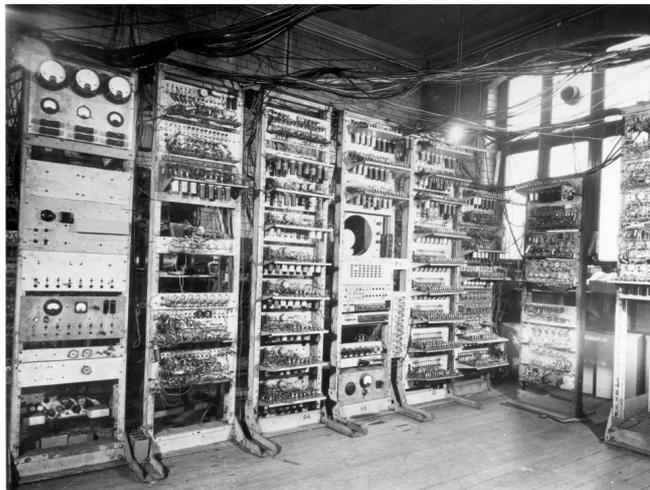
<i>University prototype</i>	<i>Date first working</i>	<i>Industrial derivative</i>	<i>Date first delivered</i>	<i>No. sold</i>	<i>Relative size</i>
SSEM (<i>Baby</i>) Mark I	June 1948	—	—	—	small
	April 1949 then Oct 1949	Ferranti Mark I Ferranti Mark I Star	Feb 1951 1952	2 7	very large very large
Meg	May 1954	Ferranti Mercury	1957	19	large
Experimental Transistor Computer	Nov 1953, then April 1955	Metropolitan Vickers MV950	1956	6	small
Atlas	1961	Ferranti Atlas 1	1962	3	very large
		Ferranti Atlas 2	1963	3	large
MU5	Oct 1974	ICL ‘P4’ (2980 family)	1975	many	large

Table 1: The five Manchester University computer projects and their industrial derivatives.

The classification in the last column of Table 1 is clearly only relative. We can add meaning by quoting in Table 2 some performance parameters for three Manchester computers that represented, respectively, the technologies of EF50 thermionic valves (tubes), OC170 germanium junction transistors and small-scale silicon ECL integrated circuits. For a short period, Atlas was considered by many to be the most powerful computer in service, Ferranti salespeople rating it as equivalent to four IBM 7094s. Up to 1,000 Atlas user-programs were being run in each 20-hour day. In 1974, MU5 could claim to be the fastest platform for block-structured languages such as Algol, though its performance for Fortran was not so sparkling.

	Peak MIP rate	Peak MFLOP rate	Primary RAM, Kbytes	Directly-addressable locations, Mbytes	No. of Modifier (index) registers	Principal Instruction Length, bits
<i>Mark I</i>	0.0006	—	0.64	1	2	20
<i>Atlas</i>	0.6	0.6	96	6	128	48
<i>MU5</i>	20	4	128	4096	1	16

Table 2: Some performance characteristics of three Manchester computers.



SSEM — the Small Scale Experimental Machine ('The Baby').

Ferranti Mercury





**Ferranti Atlas 1 at the
Science Research
Council Chilton
©Rutherford Appleton
Laboratory**

**Manchester
University MU5
under development.**



3. What ideas emerged from Manchester in the period 1948 – 1974?

It is always difficult, not to say dangerous, to attempt to attach the label ‘world first’ to any engineering idea or artefact. In truth, many hardware and software advances in computing have grown out of the experiences of several people, sometimes working on different projects at about the same time. In the early days of British computing, the National Research Development Corporation (NRDC), founded in 1949, made strenuous efforts to patent and exploit the ideas emerging from the pioneering computer groups that had appeared. By the end of 1956 NRDC held a total of 201 computer patents, including 81 (40%) originating from the University of Manchester, 43 from the National Physical Laboratory, 36 from various other government establishments under the Ministry of Supply, 16 from Ferranti Ltd., 10 from NRDC itself and the remainder from seven smaller sources. In Figure 1 we give some computing concepts and technologies that, at the time of their implementation at Manchester, were considered innovative.

Figure 1: some innovative concepts and technologies.

From SSEM and Mark I:

high-speed electronic RAM – (the Williams/Kilburn CRT store);
techniques for magnetic recording (drum): phase modulation
(*Manchester coding*);
index registers (known as B lines) and address-modification;
Mark I Autocode, an early high-level language;
primitive paging for two-level memory management.

From Meg:

early implementation of floating-point hardware.

From Atlas:

paged one-level store with hardware address-translation; large virtual
memory space;
fast parallel adder; instruction pipelining;
asynchronous operation (no clock);
memory-mapped peripherals; extracodes;
Operating System features, e.g. job control and multiprogramming;
the Compiler Compiler.

From MU5:

hardware support for block-structured languages;
compact instruction-set design;
associatively-accessed (i.e. content-addressable) fast caches, including
jump-prediction;
the controlled synchronising of unpredictable asynchronous events;
segmented virtual memory with variable-sized pages.

Some of the above ideas were soon overtaken by events. For example, by the mid-1950s magnetic core stores had replaced the CRT as the RAM of choice. On the programming side, Mark I Autocode, first available in March 1954 and therefore pre-dating Fortran by about 18 months, soon evolved into Mercury Autocode and then Atlas Autocode, by which time it looked very much like Algol. This is illustrated below by fragments of Autocode programs in Figure 2 that provide the RMS (root mean square) operation. R A Brooker took the lead in programming language development at Manchester. His Compiler Compiler led, amongst other things, to YACC (*yet another compiler compiler*) in the 1970s. Tony Brooker is the subject of an article on page 11 of this issue of *Resurrection*.

Figure 2. Coding the RMS function in Manchester autocodes.

*Note: F1 is the intrinsic function square root and the symbol * causes printing of a variable to ten decimal places on a new line.*

(a) Mark I Autocode, introduced in 1954:

```
n1 = 1
v101 = 0
2v102 = vn1 x vn1
v101 = v101 + v102
n1 = n1 + 1
j2, 100 n1
v101 = v101/100.0
*v101 = F1(v101)
```

(b) Atlas Autocode, introduced in 1964:

```
integer j; real a; array mydata(1:100)
comment calculate the RMS of real variables mydata(1 - 100)
a = 0
cycle j = 1, 1, 100
a = a + mydata(j) * mydata(j)
repeat
print(sqrt(a / 100), 5, 2)
```

The advance of semiconductor technologies and the *killer micro* have by now consigned most of the ideas in Figure 1 to the dusty archives of unremembered detail. The move away from CISC to RISC and the growth of interest in general register-set architectures hastened the process. Even the indigenous British computer industry, which thrived on the innovations of academics such as Tom Kilburn, is today no more. Nevertheless, at least two entries in Figure 1 deserve to be remembered as tokens of Manchester's many contributions in the period 1948-1974. The first is index registers and address-modification, for which the patent was filed on 22nd June 1949 in the names of F. C. Williams, M. A. H. Newman, T. Kilburn and G. C. Tootill. The second is virtual memory, as described in the mysteriously-titled paper *One level storage system*, published in April 1962 under the names of T. Kilburn, D. B. G. Edwards, M. J. Lanigan and F. H. Sumner.

The distinct Manchester innovations of index registers and virtual memory survive today in many modern computers.

Editor's note: This is a summary of a talk given by the author as part of the Digital 60 celebrations at Manchester in June 2008.

Obituary: F. J. Murray Laver

Reay Atkinson

Murray Laver belonged to that distinguished generation of Post Office engineers and scientists who moved into other government departments from 1939 onwards, there to advance significantly scientific, technical and administrative performance across the service.

From 1957 to 1965, he led that small, highly professional team in H.M. Treasury which provided the central impetus and guidance for the advances in administrative data processing in government departments over the next 15 years. The systems then developed were at the leading edge, breaking new ground and extracting high performance from a wide range of equipment operating at the limits of its capacity.

In exercising his command, Laver led by example and constructive encouragement, generating enduring loyalty in his staff. He displayed outstanding ability in resolving and presenting complex challenges. Par excellence, he was a lucid and influential communicator. This was convincingly demonstrated in 1963 when he gave a series of lectures to Under Secretaries from the principal central departments. The subject matter was common, but each session was carefully tailored to its particular audience. There never were better informed senior officials — one reason why, unlike what happened later, systems came on stream on time, within budget and performed reliably.

In 1965, Laver moved to the post of Chief Scientific Officer and director of Computer Services in the recently formed Ministry of Technology. There he played a seminal role in the re-organisation (some claim survival) of the UK's EDP industry. Mintech's initial policy (that English Electric should take over ICT) was comprehensively reversed. The future of this vital industry was thus safeguarded for some 20 years— with the odd hiccup!

Later in his career, Laver occupied a number of senior positions, including director of the National Data Processing Service, member of the Post Office Corporation and, for two terms, council member of the IEE. He enjoyed an international reputation. The honours flowed. Honorary Fellow of the British Computer Society and an Honorary Doctorate of Science conferred by Exeter University were two which he particularly appreciated, together with a 1971 CBE.

Murray Laver. A very gentle man and, to those with whom he worked, most certainly the man for all seasons.

You Can Become a Computer Programmer!

Dik Leatherdale

In *Resurrection 44*, readers were asked to complete the following series –

110 , 20 , 12 , 11 , 10 , ?

I was delighted by the interest readers have shown. The first entry arrived well before my copy of *Resurrection 44* (which says something about the postal service in rural Wales compared to that in London). David King, a student at Trinity College Carmarthen and a friend of member Huw Jones, gave the expected answer: 6.

All the numbers in the series are decimal 6. 110 is binary 6, 20 is 6 to the base 3, 20 is 6 base 4 and so on. The following readers submitted the same solution with a speed which puts my pedestrian efforts of long ago to shame –

Bob Cole	Geoff Harmer	Tim Denvir	Bob Stevenson
Stuart Fife	Tony Yule	Bob Lilley	David Yates
George Ruscoe	Chris Burton		

Several other people, however, submitted alternative solutions which merit consideration.

John Edwards noticed that $x[i] = x[i - 4] / x[i - 1]$ and ingeniously used this to get the answer 2. However, since the pattern does not repeat, this solution was deemed “tenuous, but not incorrect” by the judges.

One entrant came up with 2 (again) because, when the numbers are read as base 3 they are all factors of 2020 (decimal 60). He then honourably disqualified himself by observing that 101 (decimal 10) was missing.

Eva Lawrence interpreted the digits of each entry, naming them a, b & c (right justified). She then found that, in each given entry, $4a+3b+c=8-n$ where n is the entry number (counting from 1). The next entry, therefore, would be 2 (yet again). This is at least as correct as the anticipated solution and has a certain elegance to commend it.

So thanks are due to everybody who entered. Well done!

Once upon a time, so the story goes, somebody got 101% in the English Electric programming aptitude test. Has anybody got a copy?

Forthcoming Events

Every Tuesday at 12:00 and 14:00 Demonstrations of the replica Small-Scale Experimental Machine at Manchester Museum of Science and Industry.

Every day Guided tours and exhibitions at Bletchley Park, price £10.00, or £8.00 for concessions (children under 12, free). Exhibition of wartime code-breaking equipment and procedures, including the replica Bombe and replica Colossus, plus tours of the wartime buildings. Go to www.bletchleypark.org.uk to check details of times and special events.

Lecture Programme

17 February 2009 NWG seminar “Computers in Telephony”. Speakers David Parsons and Nigel Linge.

19 February 2009 London seminar on “EMI Computers – A Reminder”. Speakers Michael Knight & John Bradbury.

12 March 2009 London seminar “The Archaeology of Very Early Algorithms 1948-58 : Strachey’s Love Letter Generator”. Speaker David Link.

19 March 2009 NWG visit to Jodrell Bank. This visit is subject to confirmation. Please monitor the Society website for further details.

23 April 2009 London seminar “JANET – the First 25 Years”.

14 May 2009 London seminar “BBC Domesday Book Project”.

Details are subject to change. Members wishing to attend any meeting are advised to check the events page on the Society website at www.computerconservationsociety.org for final details which will be published in advance of each event. Details will also be published on the BCS website (in the BCS events calendar) and in the Events Diary columns of *Computing* and *Computer Weekly*. London meetings take place in the Director’s Suite of the Science Museum, starting at 14:30. North West Group meetings take place in the Conference Room at the Manchester Museum of Science and Industry, usually starting at 17:30; tea is served from 17:00.

Queries about London meetings should be addressed to Roger Johnson at r.johnson@bcs.org.uk, or by post to Roger at Birkbeck College, Malet Street, London WC1E 7HX. Queries about Manchester meetings should go to William Gunn at william.gunn@ntlworld.com.

Committee of the Society

Chairman: **Dr David Hartley FBCS CEng** Email: david.hartley@clare.cam.ac.uk

Vice-Chairman: **Tony Sale Hon FBCS** Email: tsale@qufaro.demon.co.uk

Secretary, Chairman DEC Working Party: **Kevin Murrell** Email: kevin@ps8.co.uk

Treasurer: **Dan Hayton** Email: Daniel@newcomen.demon.co.uk

Science Museum representative: **Dr Tilly Blyth** Email: tilly.blyth@nmsi.ac.uk

TNA representative: **David Glover** Email: david.glover@nationalarchives.gov.uk

Bletchley Park volunteers representative:

Pete Chilvers Email: pete@pchilvers.plus.com

Chairman, Elliott 803 Working Party:

John Sinclair Email: john@eurocom-solutions.co.uk

Chairman, Elliott 401 Working Party:

Arthur Rowles Email: rowles01@globalnet.co.uk

Chairman, Pegasus Working Party:

Len Hewitt MBCS Email: leonard.hewitt@ntlworld.com

Chairman, Bombe Rebuild Project:

John Harper Hon FBCS CEng MIEE Email: bombe@jharper.demon.co.uk

Chairman, Software Conservation Working Party:

Dr Dave Holdsworth CEng Hon FBCS Email: ecldh@leeds.ac.uk

Digital Archivist & Chairman, Our Computer Heritage Working Party:

Professor Simon Lavington FBCS FIEE CEng Email: lavis@essex.ac.uk

Editor, Resurrection: **Dik Leatherdale MBCS** Email: dik@leatherdale.net

Website Editor: **Alan Thomson** Email: alan.thomson@bcs.org

Archivist: **Hamish Carmichael FBCS** Email: hamishc@globalnet.co.uk

Meetings Secretary: **Dr Roger Johnson FBCS** Email: r.johnson@bcs.org.uk

Chairman, North West Group: **Tom Hinchliffe** Email: tom.h@dsl.pipex.com.

Dr David Anderson Email: cdpa@btinternet.com

Peter Barnes FBCS Email: panda.barnes@btinternet.com

Chris Burton CEng FIEE FBCS Email: cpb@envex.demon.co.uk

Professor Martin Campbell-Kelly Email: M.Campbell-Kelly@warwick.ac.uk

George Davis CEng Hon FBCS Email: georgedavis@bcs.org.uk

Peter Holland Email: peterholland@talktalk.net

Dr Doron Swade CEng FBCS MBE Email: doron.swade@blueyonder.co.uk

Point of Contact

Readers who have general queries to put to the Society should address them to the Secretary (see page 10 for postal address). Members who move house should notify Kevin Murrell of their new address to ensure that they continue to receive copies of *Resurrection*. Those who are also members of the BCS should note that the CCS membership is different from the BCS list and is therefore maintained separately.

Resurrection is the bulletin of the Computer Conservation Society. Copies of the current issue are available from the Secretary for £5.00 each.

Editor – Dik Leatherdale

Typesetting – Dik Leatherdale

Cover design -- Tony Sale

Printed by the British Computer Society

© Computer Conservation Society