

```

procedure euler (fct, sum, eps, tim); value eps, tim; integer tim;
real procedure fct; real sum, eps;
begin integer i, k, n, t; array m[0:15]; real mn, mp, ds;
i := n := t := 0; m[0] := fct(0); sum := m[0]/2;
nextterm: i := i + 1; mn := fct(i);
for k := 0 step 1 until n do
  begin mp := (mn + m[k])/2; m[k] := mn; mn := mp end means;
  begin ds := (m[n] - mn)/2; n := n + 1; m[n] := mn end accept;
  sum := sum + ds;
  if abs(ds) < eps then i := t + 1 else t := 0;
if t < tim then go to nextterm
end euler
  
```

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 (MOSI) 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,
- ◇ To develop awareness of the importance of historic computers,
- ◇ To develop expertise in the conservation and restoration of historic computers,
- ◇ To represent the interests of Computer Conservation Society members with other bodies,
- ◇ To promote the study of historic computers, their use and the history of the computer industry,
- ◇ To publish information of relevance to these objectives for the information of Computer Conservation Society members and the wider public.

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 projects 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 57 Spring 2012 Contents

The Editor Grovels (Again)	2
<i>Dik Leatherdale</i>	
Society Activity	4
News Round-Up	10
Four Years of Fun	13
<i>David Hartley</i>	
Rescuing Software from Lineprinter Listings	17
<i>David Holdsworth</i>	
The RAF Pay and Records Project 1963-4	27
<i>Michael Knight</i>	
An Archive of Early British Computing Work	32
<i>Hugh McGregor Ross</i>	
Book Review: Inventing the PC, The MCM/70 Story	34
<i>Rod Brown</i>	
Book Review: Alan Turing and His Contemporaries	35
<i>Dik Leatherdale</i>	
The Ferranti Mercury at I.C.I. – a Follow-up Note	36
<i>Tommy Thomas</i>	
David Barron : A Personal Tribute	38
<i>David Hartley</i>	
Forthcoming Events	39

The Editor Grovels (Again)

Dik Leatherdale

Resurrection Problems

Production and distribution of *Resurrection 56* did not go quite to plan. Some copies went out late and we are not completely confident that everybody got one in the end. If you've still not received your copy, you can find it at www.cs.man.ac.uk/CCS/res/pdfs/res56.pdf or contact me and I will compile a list of members to receive a special distribution.

However, those of us who did receive our copies were less than impressed by the quality of the printing. Once again the printers have failed to render accurately some of the more obscure characters in the text. When I say obscure I don't mean very obscure, for both ½ (half) and × (multiply) are within the international standard ASCII 256 character set. Frankly there is no excuse for this and I can only apologise, most of all to Tony Brooker whose article suffered in this way. Any odd “_” characters in the text should be “½” if followed by “word” or “×” otherwise. Investigations are proceeding to see if we can prevent this sort of nonsense recurring.

Programming in Schools

On page 4 we report on a speech in which the issue of teaching programming in schools was addressed. In *Resurrection 56* we reported the same subject being discussed at the MacTaggart Lecture. It is becoming an increasingly fashionable topic frequently cropping up in the media in recent weeks, not least in a long article in the winter edition of the BCS's *ITNOW*. Our friend Steve Furber has written a report *Computing in Schools* for the Royal Society. Now Michael Gove seems to have taken this accepted wisdom on board and thrown out the current ICT curriculum.

Personally I have my doubts. ICT – the processing of words, the spreading of sheets and so forth, is certainly not computer science but it is a valuable workplace skill nonetheless; perhaps the equivalent of woodwork or typing in days of yore.

Surely the days when an organisation bought a computer and then started to write bespoke programs for it are largely behind us now? And even the much-vaunted UK games industry can't create thousands of new jobs year after year? I find it difficult to believe that a shortage of programmers is a threat to the UK economy. What do you think?

Mercury Autocode

And so to happier matters. On page 36 Tommy Thomas adds some background to Tony Brooker's piece in *Resurrection 56*. Dr Thomas mentions a Mercury Autocode compiler for KDF9. But we are also aware of compilers for Atlas 1 and ICT 1900 (both called "Extended Mercury Autocode" – EMA) and the ICT 1300 under the name of "Manchester Autocode" (MAC). At Cambridge, a bright young man by the name of Hartley wrote a version for EDSAC 2 which was later ported to Titan.

And then there was CHLF – *CERN, Harwell, London and Farnborough (RAE)*. At London University, under the direction of Benedict Nixon and Alan Fairbourn, two further variants of Mercury Autocode were produced for Atlas 1 – CHLF3 and EXCHLF. We have evidence that CHLF was implemented for the Elliott 503 at RAE, for the IBM 7090 (and possibly the 709), presumably at CERN. If readers know anything about these or any further implementations, please contact the editor.

Resurrection Makeover

Keen-eyed readers will have noticed a change in the look and feel of this edition of *Resurrection*. I have long been urged by our former chairman, David Hartley to adopt a sans-serif font for *Resurrection*. I have to admit that I have the same preference but, until now I have held back. In a publication such as *Resurrection* it is I think, essential that every character should be unambiguous. Sans-serif fonts are not, on the whole, good at that sort of thing. But now, at last, I have found a font in which I (capital eye), l (lower case ell) and 1 (one) are distinct. So I offer the new presentation to readers and ask for their feedback. If you don't like it, we can revert.

At the same time, I have made some changes to the way we present website and email addresses. Hitherto, I have *italicised* them. But I also italicise other things – names of publications, some quotations and so on. So I will henceforth underline and italicise URLs and email names to make it plain to online readers where such useful information may be found.

Euphemism Corner

The Olympic ticketing website suffered a severe dose of the collywobbles in January when dealing in second-hand tickets was introduced. Within the first few hours it was "closed down for refreshment". A new term for "doesn't work"? Now why didn't I think of that? As *Resurrection 56* went to press, I should have bought the printers a pint. They must have been "in need of refreshment" don't you think?

Society Activity

The National Museum of Computing – *Dik Leatherdale*

To TNMoC at Bletchley Park in December to attend the launch of the *Domesday Touchtable*. What that? I hear you ask. Well, to commemorate the 25th anniversary of the BBC Domesday Project the information gathered together as a study of Britain in 1986 has been rescued and is in the process of being re-created. One version, *Domesday Reloaded* is, as we have previously remarked, available on the Web. But at TNMoC, a much more sophisticated presentation is available. A large, horizontally-mounted touch-driven monitor resembling a kitchen table can be simultaneously used by several visitors to find a UK location on a 1986 map and then see the information originally part of the Domesday Project with, as a point of comparison, some of the equivalent 2011 information and maps. Most impressive! Amongst the designers and implementers is one Tim Hartley – like father...

And then there were speeches. Representatives of the BBC, the National Archives and TNMoC gave of their best. Due tribute was paid to all concerned. For me, two contributions stood out:

Peter Armstrong (BBC) pointed out that the Domesday Project was an early example of what we have now come to call *crowdsourcing*. Of course, we didn't have a word for it then, but the young pupils of thousands of schools took part in the gathering of information in a way with which we are now familiar in such vehicles as *Wikipedia*. Now they are approaching middle age. Frightening, isn't it?

Howard Baker (also BBC) used the opportunity to talk about the lack of teaching of computer science in schools. He opined that the UK economy was being held back because we are failing to inspire young people to take up computer programming in the way we did 25 years ago.

ICL Archive – *Hamish Carmichael*

Four volumes of *Principles of Data Communications*, (ICL/Heinemann, mid 1980s) are on their way to the ICL Archive in the Science Museum Library.

Sundry other papers will shortly be on their way to the TNMoC Librarian (who will probably have the pleasure of recycling any duplicates).

Bletchley Park

In *Resurrection* 56 we noted a HLF grant of £4.6M. As is usual in the circumstances, the grant is dependent on matching funding from elsewhere. A number of generous donations have been made including an outstanding £550,000 from Google.

EDSAC Replica – Andrew Herbert & Chris Burton

Organization

A subset of the Board of Trustees met in late December. The registration of the project as a charity and the setting up of a bank account remain works in progress but hopefully will be complete by early in the new year.

Hermann Hauser has agreed to chair the Board of Trustees and to convene quarterly meetings to monitor the project against its charitable objectives.

A key first task once the charity is set up and the bank accounts is open will be raising an initial tranche of cash to pay for research and development. Some potential donors have been identified and introduced to the project.

Following a well-received presentation by David Hartley to the University of the Third Age a number of potential volunteers have come forward. The project manager's database now has nine names in it, in addition to those already working on the project.

Technical Investigations

Chris Burton made contact with Ernest Kaye, a pioneer from the LEO 1 project. On a recent video to mark LEO's 60th anniversary, he mentioned how he spent day after day designing circuits, so it seemed appropriate to consult him about EDSAC circuits, on which LEO was based. He gave Chris a useful reference to papers in *Electronic Engineering* from 1954 which he also found in his own archive. In turn they pointed to another paper by Lenaerts regarding EDSAC monitor displays. The former papers indicated that LEO used different flip flop circuits from EDSAC, but did confirm the unusual signal shape, with zeroes represented by small pulses. Ernest Kaye hinted that he could not remember anything about the work he did 62 years ago, but wished us luck with the replica project.

The LEO 60th anniversary videos on the Google website also gave a glimpse of a LEO store regeneration chassis and a short mercury delay line. Chris has scrutinised these in their glass cabinet in the Science Museum and, superficially, they look functionally identical with the EDSAC versions and unlikely to contribute any more information to our project.

Some further work on realising the flip flop circuit outlined in the 1948 Report resulted in a successful design which worked and was incorporated in one stage of the "Flashing Unit", what we would today call a staticisor. Further work will be needed to refine values for good tolerancing. This also included a "Reverser", what we would call an inverter - and Chris is happy with the design of that element now.

Other circuit blocks to be followed-up are the phantastron delay circuit, some form of pulse amplifier (mentioned in the early papers but with little clue as to how it was implemented) and a refinement to the Digit Pulse Generator based on one of the LEO papers.

Some further work has been done on chassis identification from photographs, but frustratingly the size of the letters on the various plates attached to individual chassis is close to the grain size of the photographic plates, so image enhancement is not viable. The next line of attack will be constructing signatures of test sockets and valve shapes and sizes in the hope of identifying the different classes of chassis.

ICT 1301 – Rod Brown

The ICT 1301 project is still running in December as the warm weather is still allowing us to turn the machine on. However work is now limited to just the data transfer interface, which is currently suffering from a rogue timing problem where it loads the FIFO chip, but cannot load the serial monitor interface correctly. Suspicion falls on the UART chip for which we are awaiting a replacement chip from USA. Yes, we really are extending a nearly 50 year old machine with 25 year old chips!

Further visits from members of the U3A are planned for 2012 as are the events which mark the 50th year of this machine. 2011 has provided us with a large number of new contacts and lots of donated hardware and manuals.

When the dropping temperature forces us to close down we will be placing the first public release of a simulator on line. This simulator is only about 87% fully functional but it is being provided due to demand and we have several alpha version testers who wish to "play" with what we have produced so far. The current build runs program loops and loads simulated program card packs.

Progress this year has been slow but steady and, unlike earlier years, we end the year without losing any percentage of progress towards our overall target.

As usual all updates are online at www.ict1301.co.uk/1301ccsx.htm.

IEEE Annals of the History of Computing– *Hamish Carmichael*

For the past ten years the Computer Conservation Society has run a corporate subscription to the Annals, allowing our members to receive copies at a considerable discount. In order to prevent a rise in price it would be useful if we could increase the number of subscribing members. Any member who would like to join the scheme should contact Hamish Carmichael.

Hamish still has a few spare copies of Volume 27 Number 3 of the Annals, which under the heading of Historical Reconstructions has, among others, excellent articles on *Replicating the Manchester Baby*, *The Rebuilding of Colossus* and *The Construction of Charles Babbage's Difference Engine*. Any member who would like one of these should contact Hamish Carmichael; first come first served.

Harwell Dekatron – *Johan Iversen*

Delwyn and I have started looking at the power supply. We removed the transformer and stabilizer units from the Witch rack, connected the transformer unit to a Variac and measured the output voltages. They were found to be quite high but we assumed this was because there was no load. Otherwise no other apparent problems were seen. The stabilizer unit was connected to the transformer unit, again with no load. Winding up on a Variac we discovered excessively high output voltages at a mains input voltage around 180V and a glowing hot anode. Even when a load was added there was little change in the output, however after some further tests it was concluded that stabilizer unit was functioning correctly. The transformer unit therefore required further investigation. After some inspection it was found that after its rewire a couple of connections to one of the transformers were wrong. During the correction of these connections oil was discovered on the transformer terminals, further investigation showed a large amount of oil around the transformer along with a small pool on the bench. The transformer was removed and it was discovered that the oil was leaking from along the seam of the case join. The good news is that further tests, including insulation tests, showed that the secondary outputs under load were within spec. If we can find a way to repair the leak the transformer should be fine. However another problem has come to light. In tracing the actual circuit and comparing it to the schematics a discrepancy was found which could be the cause of the high voltage that has been observed. We are not sure when this mod was done.

Elliott Project – Terry Froggatt

The Elliott 803 has had several problems recently. The floating point overflow lamp lit, when running a program which did not use floating point. Initially a suspect OC84 transistor was replaced, but when the fault reappeared some weeks later the true culprit was found to be a supposedly 10ohm resistor measuring 15ohms.

Following a cold night, the core store took about 1hr to become reliable. On another occasion the 803 turned itself off after about 30 seconds. Investigations found a blown 15A fuse on one of the +10V supplies.

A transient fault with the initial instructions logic has not reoccurred. The long standing fault with the tape punch logic has not yet been isolated.

Plans are being made to externally test the plotter interface before putting the board into the 803.

Recently an 803 cabinet was collected from a farm house cellar near Banbury. It used to be a power cabinet but had been converted into a gun cabinet! The plan is to clean it up, add some internal shelves and to use it to hold the tape library.

The extra store module for the Elliott 903 has been cleaned up and tested. It had three faulty cards, two of which are of no consequence in the present configuration. The third card has temporarily been replaced by a card from an incomplete 903. Another 903 desk in good condition was recently donated by a CCS member and this has been used to house the store module.

The two-desk system is now on public display with a total of 16K of 18-bit words. It is normally demonstrated running BASIC, on a DEC VT220 terminal which has the correct 20ma current-loop interface (and uses no consumables) but is of the wrong era. As an alternative, a Teletype (of the right era but with the wrong interface) has now been interfaced to the system (using an RS232/20ma dongle). Work continues on the engineering display panel.

Saturday 5th November was a particularly interesting day at TNMoC, when three past Elliott service engineers visited simultaneously. One of them tweaked the peripherals on both the 803 and the 903, another has some knowledge of the 905 motorway signalling system and might be able to help with the TNMoC 905.

Finally, TNMoC has just acquired a handful of paper-tape punches in good working order, from MoD Southwick Park. Work has started on converting some of them to run on 240v and to interface with Elliott paper-tape stations.

Ferranti Pegasus – *Rod Brown & Others*

At last some positive news about Pegasus! It has been agreed that the proposal of an external consultant to safety test and certify the machine is not viable and does not represent value for money. That may not sound like good news, but it is. The decision clears the way to explore other avenues. Other avenues are being explored with some energy and it is hoped that we may see some progress in the near future.

The long-running saga of role definitions is also almost at an end with the introduction of a specific role of “presenter” whose job it will be to talk to visitors without the distractions of actually having to work with Pegasus at the same time.

Analytical Engine – *Doron Swade*

The Science Museum has completed the digitisation of the Babbage technical archive. This is a landmark accomplishment in the history of this material first loaned by Babbage’s son to the Library in 1878. Plan 28 and The Science Museum are currently working through details of an agreement to enable us to have early access to the digitised images and we expect that process to be finalised by the end of the year. Analysis of the designs and specification of what is to be built is the next stage of the project. Once the material is accessible to us detailed research will begin in earnest.

Bombe – *John Harper*

Last year I reported that we were in touch with the Heinz Nixdorf Museum, Paderborn. The original proposals did not come to fruition, but instead we committed to produce a dummy checking machine that is almost 100% correct visually but with the internal mechanisms based on modern microprocessors. To all intents and purposes the machine operates exactly the same as the real item. This unit is now complete. In addition we have produced demonstration instructions. The unit will be displayed and demonstrated in Paderborn for a month or so as part of their Turing Anniversary programme.

News Round-Up

The big news in this edition of *Resurrection* is that the Museum of Science and Industry at Manchester (MOSI) is to become part of the National Museums of Science and Industry (NMSI), the parent organisation of one of our founding fathers, London's Science Museum. The move follows the ending of central government funding for museums which are not designated as "National". MOSI's director, Tony Hill has resigned. During our recent visit to MOSI, we were delighted to meet Tony and were very impressed by his willingness to take our concerns on board. We wish him well for the future. More detail in the *2nd December edition of the Manchester Evening News*.

101010101

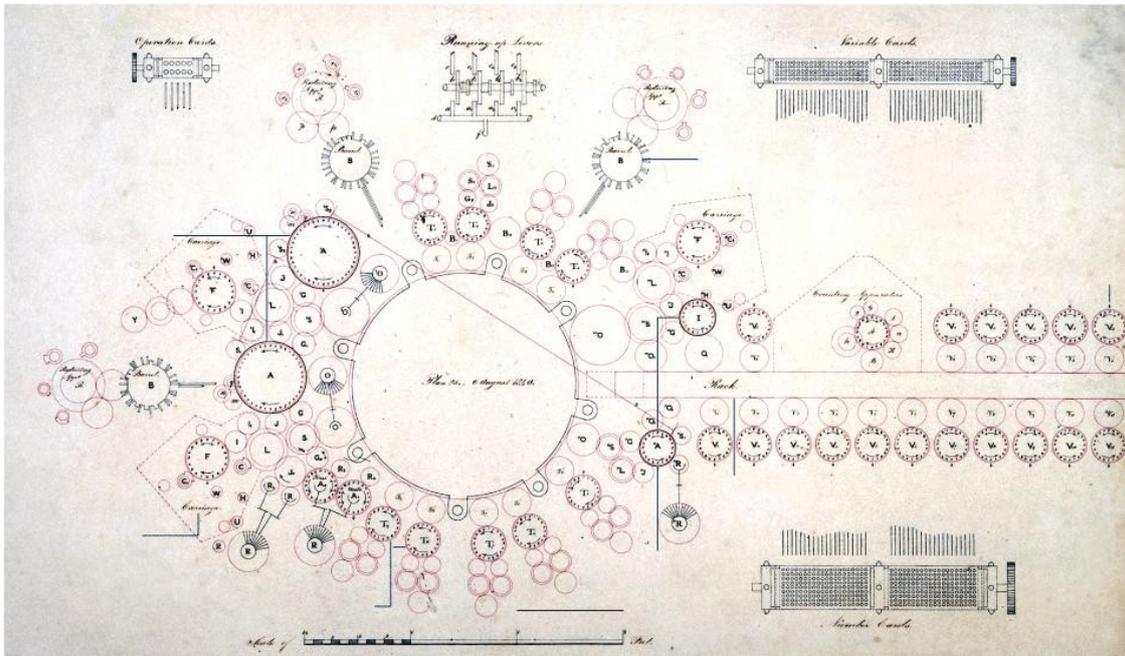
On 17th November 1951 LEO 1 ran its first productive work – bakery valuation. The 60th anniversary of the event was marked by a celebratory lunch organised by our friends the LEO Society and hosted by Frank Land. Many of the prominent LEO veterans were present,



together with CCS Chair Rachel Burnett, David Hartley and Peter Barron, Google UK's PR Chief. Professor Land paid due tribute to all the LEO pioneers and acknowledged the part played by the late Sir Maurice Wilkes. Peter Barron expressed both his surprise on learning of the circumstances of LEO's creation and his admiration for the achievement. "The fact that Lyon's achievement turned out to be true," he said, "must surely be one of the most remarkable facts of computer history."

Earlier in the day, there was an interview on BBC Radio 4's *Today Programme* in which Frank Land held forth on how Leo came into being and what it was like to pioneer what we later came to know as "data processing". The interview can be heard at news.bbc.co.uk/today/hi/today/newsid_9637000/9637100.stm.

101010101



News of the plan to construct Babbage's Analytical Engine has reach as far afield as the *New York Times*. An article describing the project can be found [here](#) or go to www.nytimes.com and search for "Babbage". The article was later reprinted in *The Observer*.

101010101

Alan Turing was the subject of the *My Hero* feature in *The Guardian* in November. Written by one of Turing's friends, it can be found at www.guardian.co.uk/books/2011/nov/11/alan-turing-my-hero-alan-garner.

101010101

Unlikely as it sounds, it is to the *Radio Times* that we owe the intelligence that a new film about Alan Turing is planned, based on Andrews Hodge's "magisterial" biography. The star is to be current heartthrob, Leonardo DiCaprio.

But we have been here before, have we not? In the 1980s, Hugh Whitmore's *Breaking the Code* starring Derek Jacobi and based on the same source was triumphantly received on its transfer to Broadway. It came to the attention of a Hollywood producer. The playwright recalls being asked for some changes – "I don't want this guy to be a faggot and for God's sake cut out all the mathematics." It is to Whitmore's great credit that the project proceeded no further.

101010101

News has just reached us that the Bletchley Park Trust has appointed a new CEO – one Iain Standen, recently retired as a Colonel in the Royal Corps of Signals.

We hear that David Link has unearthed from Oxford's Bodleian Library Christopher Strachey's famous draughts program written in 1952 and has succeeded in making it run again on his Mark 1 emulator. It is, however, as yet unfinished. He writes –

"I would be grateful for any recollections on the program. Perhaps some of you played against it in the 1950s or you know somebody who did?"

I am still struggling to understand the way in which Strachey's program detects if the user has input information. The code is:

- set accumulator to some high number*
- repeatedly count it down by subtracting 1*
- if accumulator goes negative, switch M and L, the two sides of the accumulator*
- test if accumulator negative. If it isn't, user has hit "KAC" – key accumulator clear.*

I would have thought that both methods – counting down and KAC – would set the accumulator to zero. In the next round of subtracting 1 both M and L would be filled with 1s, independent of the way it was set to zero ... But obviously, this isn't true.

Any ideas highly appreciated!"

David Link can be contacted at david@khm.de.

101010101

On 1st April 1976 (no, I'm not making this up), three young men signed legal documents bringing the Apple company into being. Steve Wozniak and the late Steve Jobs found fame and fortune, but the third man, one Ronald Wayne, remains deeply obscure. After 11 days he seems to have got cold feet and resigned, hence his continued low profile. His initial investment of \$800, some 10% of the capital, was refunded. More legal papers were drawn up to document the change. Wayne's copies of the founding documents were sold for "several thousand dollars" in 1994 and in December 2011 were resold at Sotheby's for a smidgen short of \$1,600,000. Details can be found at www.bbc.co.uk/news/technology-16170953. Amazing what some people spend their money on, isn't it? Never throw anything away!

101010101

Four Years of Fun

David Hartley

To say that it has been an honour to serve as chair of the CCS these past four years is something of a cliché but nonetheless true. Though to say that it has been fun is much nearer the mark. The cliché is more relevant when one has served in the upper echelons of an organisation such as the BCS, where fun is seldom the right word when one is submerged in councils, boards and committees.

The tradition of the CCS to elect as chair past BCS presidents is, to my mind, a delightful one. One is re-engaged into the BCS at more or less the bottom of a long hierarchy and, for me, it has been a pleasure



Formal Mode

to serve and discover a highly successful body involving many excellent people who blend together with a sense of mutual respect and form great friendships. The regular gathering at the *Hoop and Toy* in South Kensington after the monthly lecture at the Science Museum is testament to this.

Perhaps of its nature, being involved with the past and the history of computing, the committee is mainly composed of those who are retired. Those who are not retired will comment that we have plenty of free time to play with computer restoration and the like. Looking at the CCS committee, I have never seen such a hard working and fully engaged group. There is, of course, one outstanding exception in that the person probably most hard working is our Secretary, Kevin Murrell. Kevin supports the committee, keeps track of CCS membership generally and is also a director and trustee of The National Museum of Computing. He does all this as a volunteer and also has the proverbial day job, running his own software company.

My purpose in writing this article is to highlight what I admire about CCS and to pay tribute to the members. It is difficult not to mention names (I

just have done) and it is inevitable that I shall in what follows. But the only way to describe the achievements of the CCS is to mention people even though there is a danger of omitting to say what should be said as well as saying what should not be. But, nothing ventured – nothing gained, here goes; I was ever one for not being wholly tactful.

One of my first reactions was to wonder at the size of the CCS Committee. At the last count it had 32 members each with a specific responsibility. My natural reaction was to wonder whether such a large body should be structured. But beyond a little light pruning, past BCS experience had taught me the dangers of piling bodies on sub-bodies on sub-sub-bodies. There was certainly a problem allowing everyone to have their say, when business started at 11am and finished at 1pm; business had to be completed in something of a rush. So with Kevin's help, I bullied everyone to get their reports written and circulated in advance, verbal reporting being confined to last minute developments or urgent matters. The chairman would be seen to glower at those who failed to keep the rule.

So what are the highlights of my four years? Firstly we were able to survive the rather sudden resignation of Nicholas Enticknap as editor of *Resurrection* after 18 years and 42 issues. Nick had served since the CCS was founded and to replace him would be a challenge. But before we had time to make enquiries for a successor, Dik Leatherdale had raised his head above the parapet. It was clear he really wanted to do the job and, after a rather informal interview, we invited him to serve for a probationary year. Needless to say, the probation was as much to see whether his obvious enthusiasm would be sustained rather than any doubt about technical skills. The enthusiasm remains and *Resurrection* is in very good hands.

A notable achievement in which I take a little pride was the enticement of the ICT 1301 into the CCS fold. This project, enthusiastically run by Rod Brown (in a barn in deepest Kent), had been considered for CCS membership before I arrived. The committee had not unreasonably taken the view that there should be some criteria for accepting new projects. But it seemed no one had the energy to make proposals for such criteria and the 1301 remained apparently forgotten or ignored. I decided this was no way to treat a potential new family member and visited the project accompanied by Hamish Carmichael. At the next committee meeting I asked whether anyone had any reason not to invite them to join us. The silence which followed was sufficient to resolve the issue. And we gained not only an excellent project for the CCS portfolio, but also an excellent committee member in the shape of Rod Brown.

Not so much a highlight – more a severe difficulty, has been the problem with Pegasus at the Science Museum. Following what turned out to be a rather minor incident in the machine's power supplies, which caused more smoke to be emitted than real fire, the authorities decreed a cessation of operations while a lengthy investigation took place in search of non-existent asbestos. There has been much recent reaction against the excesses of health and safety practices and we are told that to be reactionary is unhelpful. Maybe so, but after two and a half years Pegasus continues to sit forlornly in the second-floor gallery of the Museum like, in the words of a much missed former colleague, a proverbial lump of Carrara marble – white, perfect and immobile. The good side of the issue has been the realisation of the need for formal certification and training of volunteers who look after the fruits of CCS projects. Much work has gone into creating definitions and specifications. We still refer to Pegasus as the oldest working computer in the world; unfortunately it hasn't worked for over 2½ years and the designation is wearing a little thin.

There is another basic issue that the Pegasus saga has highlighted. Professional museum curators on the one hand and the essentially more amateur enthusiasts of computer conservation and restoration on the other hold diametrically opposing views. The former are concerned to preserve both for history and education the authenticity of artefacts, while the latter believe in the educational value of having things working. Operational artefacts have to be maintained and maintenance lessens authenticity. The CCS subscribes to the view that there is nothing so boring as a dead computer. It is good report, however, that the CCS and the Science Museum were able to resolve the dilemma in the context of the future of Pegasus

One unresolved major area is archiving. We accumulate in a rather opportunistic way many old documents. Some are original manuscripts that have historical significance, others are simply piles of old computer manuals, ejected from the attics of the deceased by widows and dumped on whoever will find storage space. What should be preserved as important and sometimes rare historical artefacts archived and catalogued? What should simply be thrown away? The cataloguing in itself can be a mammoth task and the decision as what to keep and where and how to make it accessible requires skill and resource. Hamish Carmichael, for example, has completed an heroic project to catalogue the whole of the ICL Archive (which, of course, goes back to the myriad of former companies that formed this once dominant British computer company whose future has now disappeared into the even larger Fujitsu). Simon Lavington has been cataloguing essential documents relating to early

British computers and work has started to make this information accessible via the web. The options for collection, storage, accessibility and presentation are enormous, as is the total space of the subject whether from the angle of technology, people or applications. To mix the metaphors we need to do more than scratch the surface. We need to understand the landscape, discover its environment and find techniques that will enable long-term educational value to be obtained – before there is no one left who remembers how things were in the past.

I was fortunate in my four years to pay two useful and important visits to the Computer History Museum in California. The first was in 2007 to celebrate, with Doron Swade, the inauguration of the second copy of the Babbage Difference Engine, the rebuild of which was pioneered by Doron when he was at the Science Museum. At the time CHM was still developing their new galleries following the transfer of the collections from the previous museum in Boston. The second visit in 2011 was to go, with Kevin Murrell, to see the completed galleries as well as to talk about the recently established CCS project to rebuild the Cambridge EDSAC. The occasion was also used by me to give the first of four public tributes to the late Sir Maurice Wilkes who died in 2010. Maurice had been an enthusiastic supporter of the CCS since its foundation in 1989.

Which leads me finally to the sad death of Tony Sale in the last month of my chairmanship. I don't need to repeat the things I said in an earlier issue of *Resurrection*, except to note that he deserves to be known and honoured as a founder of CCS, who through his tenacity and brilliant engineering skills created so much of what the CCS is and stands for today.



Informal Mode - Spot the Common Factor

As we completed my last CCS Committee meeting as chair, a number of colleagues said some rather kind things. In addition I was invited to remain a member of the committee – complete and sudden severance would have been difficult to bear. I was described as one who always spoke his mind but did so with both warmth and vigour. That was a nice way of describing a lot of fun.

Rescuing Software from Lineprinter Listings

David Holdsworth

A basic tenet of the CCS's software conservation activity is that software is only truly conserved when it can be run on current hardware and has the prospect of being run on hardware of the foreseeable future. The preservation of software which is available only as a lineprinter listing can easily seem to be impossible in the terms expressed above. Although OCR can sometimes seem to have miraculous capabilities, faced with a 40 year old lineprinter listing, current OCR software produces at best a pale shadow of the original and the prospects for achieving the accuracy necessary to produce working software would seem to be poor, especially as such software is likely to be written in assembly language. This paper discusses techniques for bringing such software to life in the light of two successful examples, both of them software for KDF9, but written in different assembly languages and rescued using different techniques for accurate copy-typing of several thousand lines of code.

KDF9 was a workhorse machine in nine universities and around a dozen other institutions, at a time when systems software tended to be written in assembly languages, although end-users were increasingly using emerging high level languages. One of KDF9's early claims to fame was its Whetstone ALGOL system, often known to its users as WALGOL. The KDF9's other ALGOL 60 compilation system, Kidsgrove ALGOL, was similarly nick-named KALGOL. In the spring of 2009, a lineprinter listing of WALGOL came into the possession of Brian Wichmann (BAW). There was already some activity in emulation of the KDF9 by Bill Findlay, and BAW raised the question of the possibility of getting WALGOL working again.

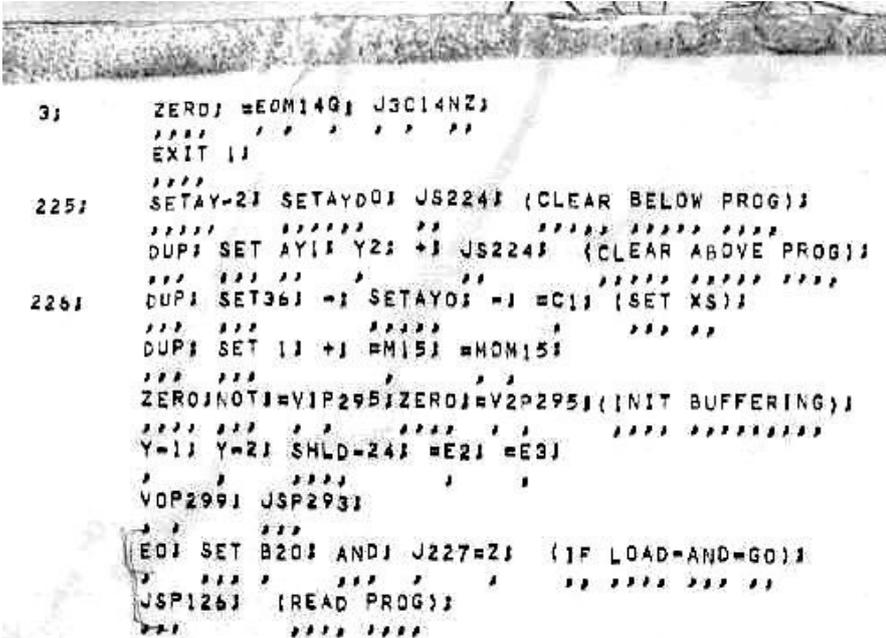
It is this listing of Whetstone ALGOL that forms our first example and is written in KDF9 Usercode – the official assembly language. The second example, of which more later, is Leeds University's assembler for a new KDF9 assembly language, KAL4, which is itself written in KAL4. We now have the ability to execute each of these in a wide variety of current software environments.

Lessons learnt

- Copy-typing can deliver accurate code
- It is best to have typists who have some recollection of the language being typed.
- Do not burden the typists with arbitrary rules
- Do not lose heart
- Early emulation motivates typing
- Make software tools to cut down on drudgery — especially an emulator with copious diagnostic output.
- Keep in mind the abstraction of the true digital original, especially when dealing with a language processor written in its own language.

Whetstone ALGOL

The WALGOL system is divided into three parts, the ALGOL translator, the assembler for Usercode bodies and the runtime system. The second of these only arrived later and the system ran



```
3;      ZERO; =EOM14Q; J3C14NZ;
      ****
      EXIT 1;
      ****
225;    SETAY-2; SETAYD0; JS224; (CLEAR BELOW PROG);
      ****
      DUP; SET AY; YZ; +; JS224; (CLEAR ABOVE PROG);
      ****
226;    DUP; SET36; -; SETAY0; -; =C; (SET XS);
      ****
      DUP; SET 1; +; =M15; =MOM15;
      ****
      ZERO; NOT; =VIP295; ZERO; =V2P295; (INIT BUFFERING);
      ****
      Y-1; Y-2; SHLD-24; =E2; =E3;
      ****
      VOP299; JSP293;
      ****
      EO; SET B20; AND; J227=Z; (IF LOAD=AND=GO);
      ****
      JSP126; (READ PROG);
      ****
```

Fig.1 - Part of the WALGOL listing

without the capability for Usercode bodies for several years. What we started with was the ALGOL translator which ran to 84 pages and the runtime system (known as the controller in its heyday) which ran to 153 pages. This is not as terrifying as one might at first think, as the format of the printer listing used 2 lines for each line of source text (something called UPDATER format). Nonetheless, this represented 2723 lines of Lawford Russell's translator and 5016 lines for Brian Randell's controller, a total not far short of 8000 lines of quite densely packed assembly code with very few comments. Both KDF9 assembly languages allowed multiple instructions on a line. The total amount of code is just over 50kbytes and with KDF9 instructions having variable lengths of 1, 2 or 3 bytes, we were

probably dealing with about 25,000 assembly language instructions, a total of over 200,000 characters.

OCR was quickly seen as infeasible (see fig.1). The bulk of the work was done by a team of 5 people, 3 typists (GT, RMcL, BAW) and one collator (DH) produced the machine readable source text and a new Usercode assembler, while the sixth person (WF) produced the emulator.

Modus operandi

BAW photographed all of the pages and despatched them in batches of 10 to the typists. Two of us live at opposite corners of England, two of us live at opposite corners of Scotland, while the fifth member of our team lives in the USA. We have never met as a team and almost all communication has been by email, supplemented by very occasional use of the telephone.

Every page was typed independently by two people and the typescripts emailed to the collator, who wrote software to pre-process the text ahead of the `diff` command (actually the GNU version). He also wrote a Usercode parser, editing the on-line definition of the order code into `yacc` input. All exceptions either flagged by `diff` or the parser were checked usually against the photographs and an error free version of the page produced by editing the better of the two raw typescripts.

Then all that remains was to extend the parser to be an assembler, the output of which could be fed into the emulator. That in essence, is what happened but with several lessons learnt along the way as various hurdles were surmounted.

Character codes

The KDF9 predates ASCII, which in any case was not universally loved by all collaborators. As a result there was a variety of opinion as to how the non-ASCII characters (`÷ × ≠ ≥ ≤ ₁₀ ↑`) should be represented. All of them feature in the source code, although the last one appears only in comments.

Motivation is vital in an exercise such as this, so we minimised the rules for typists and encouraged individuals to use whatever representations they preferred. The collation software accepted all of the different choices and produced a canonical form before the use of `diff`. We only had one false positive where two equally valid lines were still different after the canonicalisation.

Error rates

Only a very few pages were typed where one of the versions was error free. When the first version of a page arrived, it was run through the Usercode parser where some errors were often found. But it was the comparison of duplicate type-scripts that was the most effective detector of errors. Nonetheless, we have detected two places where both typists made the same mistake, in each case the replacing of a letter **I** by the digit **1**. It is an annoying feature of Usercode that such typing errors can quite often not lead to syntax errors. The first of these was detected when we first tried to execute the code and was in the 23rd instruction to be obeyed. Consequently, we found it in the early stages while emulating with maximum diagnostics. The second such error survived much longer and was only detected by a detailed study of the code, after getting runtime errors reported by the controller.

Early in the process we noticed a non-sequitur in the code. It transpired that the photographer had turned two folds at once and missed pages 13 and 14. Unlucky one might say.

The partially complete source was fed into the assembler each time a new page was added and missing labels were ignored. There was just one label in the translator that was unreadable and, when the complete source was assembled, only one label was missing. Though it was difficult to interpret the splodge on the listing as the missing label we made that assumption. Subsequent experience would indicate that the assumption was valid.

Losing heart

On 22nd April 2009, BAW emailed the team with "I must say that I am concerned about getting it working since the code seems very difficult to understand." At this stage we were only about ¼ the way through the translator, but the Usercode parser had blossomed into an assembler which was producing a convincing looking binary program. I personally needed to see something working to keep up motivation and the emulator project was some way off. I knocked up a rough and ready emulator using the techniques from the ICL1900 George3 work. All instructions are initially implemented as illegal then you start the execution. As each instruction is flagged illegal you implement its emulation. It was not long before discovering the duplicated mistake in the 23rd instruction, which rather suggested that there was substance to Brian's email. However, the emulator was soon obeying hundreds of instructions.

The prospect of eventual success, however distant, was wonderfully motivating. BAW disciplined himself to do one page per day and, before

too long, we had the translator (about a third of the whole) complete. In an amazingly short time, it would translate ALGOL programs using my crude emulator, at least until any real numbers appeared in the ALGOL.

I doubt that without this emulation success, we would ever have summoned up the motivation to copy-type the controller which is twice the size of the translator. BAW never faltered on his one-page-per-day commitment and delivered the final piece (a duplicate copy of page 100) in an email time-stamped 0749 1 Dec 2009.

Getting it working

My rough and ready emulator was not really up to running non-trivial programs. Brian Randell observed that it was the first time that WALGOL had run "Hello World", as its rise to popularity as a köan for compilers came rather later.

WF's emulator arrived in early 2010 and before long we were successfully running old ALGOL 60 acceptance tests and benchmarks that had lain dormant in Brian Wichmann's possession for some years. As several of us had not used ALGOL 60 since the 1960s, our fluency with the language was somewhat questionable and it is interesting to see just how a 1960s ALGOL compiler questions that lack of fluency. Most of today's compilers use an LALR1 parser (e.g. `yacc`) and the reporting of syntax errors looks pretty much the same whether using `gcc`, `javac` or many others. WALGOL uses an operator precedence parser. Not only are the error reports quite different in the way that they point at the error but the line counts do not include the blank lines. In those days, the assumption was that you were counting lines by hand in a paper listing.

To get it working on your machine, just look at sw.ccs.bcs.org/KDF9.

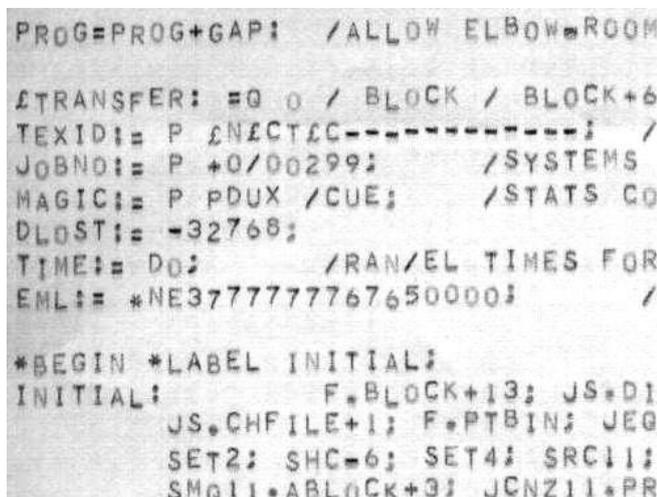
KAL4 – a KDF9 assembler

Some months ago, I was given a lineprinter listing of the KAL4 assembler. This assembly language for KDF9 was implemented as a student project at Leeds by John Thomason. The motivation came from the woeful inadequacies of KDF9 Usercode as a vehicle for writing and maintaining software. We identified three previous implementations of assembly language for KDF9, so ours was KDF9 Assembly Language 4. Such was the success of the project that the language was adopted for systems programming on the Leeds KDF9. It was certainly used for writing our job scheduler (JOBORGANISER) and our FORTRAN compiler. This compiler generated KAL4, so the assembler was in frequent use.

The only KAL4 program currently in our possession is the KAL4 assembler. For many weeks I pondered the prospect of doing the bootstrap, before hitting on a technique for casting KAL4 into an LALR1 grammar, by slightly quirky lexical analysis. Thus my Usercode assembler could be adapted as a KAL4 assembler, once again using **yacc** to generate the parser. Now that there was the prospect of running the assembler, there was the motivation for trying to resurrect it.

Modus operandi

The KAL4 listing was no more amenable to OCR than the WALGOL listing – actually, less so (see Fig.2). It was listed in the POST format which uses less paper, as there is only one line of print per line of text, but the case of letters is lost. So the 26 pages of listing are equivalent to 52 pages in the format of



```
PROG=PROG+GAP; /ALLOW ELBOW=ROOM
&TRANSFER: =0 0 / BLOCK / BLOCK+6
TEXTID: = P &NECTAC-----; /
JOBNO: = P +0/00299; /SYSTEMS
MAGIC: = P PDUX /CUE; /STATS CO
DLOST: = -32768;
TIME: = D0; /RAN/EL TIMES FOR
EML: = *NE377777767650000; /
*BEGIN *LABEL INITIAL;
INITIAL: F.BLOCK+13; JS=D1
JS.CHFILE+1; F.PTBIN; JEQ
SET2; SHC=6; SET4; SRC11;
SMQ11+ABLOCK+3; JCNZ11+PR
```

Fig.2 - Part of the KAL4 listing

the WALGOL listing. It needed typists who knew the language and conventions for use of upper and lower case and who were also motivated. Only one such person was identified, but I decided to go ahead without the prospect of duplicate typing and to rely on emulation to tease out the errors. After all, that had been the technique that found the two errors that had been duplicated in the WALGOL copy-typing.

Error rates

Unfortunately I did not initially keep track of error rates. The earliest detection of errors was by the assembler, but only saw those that caused syntax errors, not the omitted instructions, omitted lines, any error that does not violate the syntax. I did find an early version after the syntax errors were eradicated and compared it against the version that can bootstrap itself. There were 52 errors, an average of two per page, one every 30 lines. There were, perhaps, twice that number originally.

Getting it working

KAL4 has mnemonic labels and these were a great help in understanding the code, even though there were not many more comments than was the case with WALGOL. It is more obvious that **JS.oubs**; outputs an ALGOL basic symbol than **JS14P295**; does. The emulator can be run in a

diagnostic mode in which there is a line of output for each KDF9 instruction obeyed. Furthermore, this mode can be turned on at a chosen point in the execution. Also, my new assembler gives a very verbose listing showing all the generated code against the source text.

So, my technique was to set the emulator off with the newly assembled assembler and no diagnostics. An error report would ensue, giving an address of failure. I can run the emulator just reporting subroutine exits and this trace was very useful in deciding whereabouts things started to go wrong — especially in those few cases where my typos caused it to loop indefinitely. I would then rerun so as to turn on maximum diagnostics just before the supposed error, giving me a list of instructions as they were obeyed.

I eventually discovered that the best way to make progress was to follow through the original source text using the scan of the printer listing (or the listing itself), after first using the assembler's output to locate the page in the listing. I inserted comments into the source text which were hot links to the scanned source text page in the assembler output. (At this stage all assembly is using my newly written assembler. The original KAL4 assembler does not yet work. That is actually the point of the exercise.) It was now possible to read instruction-by-instruction against the original and when I hit the typo it would jump out that the instruction shown on the log was wrong. I soon came to trust certain frequently called subroutines (e.g. fetch next symbol) and could skip by their traces by searching the log (viewed using an editor) for the exit address.

It was not long before the emulated assembler tried to read the source text from the filestore. The filestore API was just a collection of 640 word blocks, each with its unique address, block zero being the root of the index. These blocks were read by a system call known as OUT32. Any program could read any block and hence any file, but a program did need special status to be allowed to write to the disc.

The assembler's desire to read the source text first manifested itself as a system call to read block zero (OUT32 in KDF9speak). By a great stroke of luck the filename of the desired source text was held in the KDF9 central registers at this point (its famous nesting store). So, the emulator was enhanced to implement OUT32 for block zero by delivering an index for a filestore holding only one file, the one desired. It can be argued that this fails to verify the code for searching the index, especially the situation where the desired file does not exist. However, I got the emulated block format wrong at first, and so tested this situation quite well — and even tested the situation of a corrupt master file index.

To begin with the assembler was only given tiny KAL4 programs to deal with.

Character codes and Filestore

Although the same non-ASCII characters occur in KAL4 as in Usercode, the KAL4 implementation only operated within the Eldon2 and PROMPT world and the source code was read from the filestore, which held all text as ALGOL Basic Symbols (ABSs) in its own 8 bit code. So the digital original can be regarded as the file of ABSs, of which the lineprinter listing was but one possible manifestation. Had the same file been listed on a FlexoWriter, the appearance would have been different in several respects from the lineprinter listing and different again from any listing obtained on one of our Eldon2 multi-access teletypes.

For input to my new assembler, I needed an ASCII file, but once I was generating a binary program which then ran under emulation, I needed to feed the assembler with the file of ABSs that it was written to process.

I added emulation of OUT32 for non-zero addresses in which the contents of the block were constructed from the source text, converting the ASCII to KDF9 ABS code on the fly. A later alternative constructed the 640 word blocks using a separate program run before running the assembler.

Bootstrap

In due course, we got to the point where the little tiny KAL4 programs were assembled correctly and could easily be run using the assembler's load-and-go option.

Time to jump in at the deep end and try self-assembly.

The technique of reading the diagnostic log against the actual listing continued to tease out errors. Sometimes, it was necessary to create a tiny program containing an example of the construct that had flagged an error.

In my new assembler, I had not quite implemented all the language, particularly the many and varied forms of constants. This necessitated some places in which the copy-typed text constructed the same constant values, but using the language differently. Thus the source text for self-assembly eventually became different from the source text processed by the new assembler. However, when errors were found they were corrected in both versions.

At this stage we have a machine-readable source text in which the ALGOL basic symbols are represented in ASCII and a KDF9 binary program assembled from this source text which is capable of producing itself from

this source text. It may have errors in code which processes parts of KAL4 which do not occur in itself. It is not surprising that a student implementing his own language by bootstrap should wish to show off the features of KAL4, especially perhaps those which are not available in Usercode. This did mean that self-assembly was a really good test case, but the assembler itself contains no floating point constants.

Because of the imperfect implementation of the PROMPT/Eldon2 filestore, a kludge was put into the emulator to extract the binary image in a form compatible with our two emulators.

Time to explore the parts of the language that were not used in the assembler itself, primarily numeric constants. The very simplest numbers with decimal points produced mayhem and pointed to an area of the assembler with complex use of KDF9's nesting store. Right in the middle of this sequence of impenetrable code was one of KDF9's most mysterious instructions, $\div R$, or its KAL4 synonym DIVR, a divide instruction for multiple length division. Both writers of emulators had left this instruction as unimplemented, in the belief that it was not understood by any living person and unlikely to occur in any of the software that we possessed. After several false starts, we now have this instruction implemented with great confidence, because the KAL4 assembler now tackles numeric constants with aplomb.

Conclusions

Firstly, we really can take a printer listing and turn it into machine readable code which is accurate enough to run to the point that we know of no bugs in it. We did find a bug in Whetstone ALGOL that English Electric had fixed in later versions. We edited the correction into our source text, but with a comment to that effect.

Anyone embarking on such enterprises as these must expect to have to make their own software tools, but using modern languages and compilers coupled with fast personal machines while processing decades old software underlines quite spectacularly the improvements in software production that have gone alongside the awesome hardware progress in computing.

So what have we preserved? To what extent have we only produced replicas?

In digital information processing the replicas are perfect copies. The backup copies are perfect substitutes for the originals. Our source texts look a lot like the printer listings from which we started and in any case the printer listings were only replicas/representations/renderings of files

on disc or magnetic tape. I believe that we can claim that our binary programs are functionally equivalent to the originals and extremely close (possibly identical) to the original bit-stream. Our source texts in ISO-Latin1 look like the originals when printed or displayed on screen. We do have code which generates HTML which looks like the printout from a KDF9 FlexoWriter when read by any browser. Our source texts can be browsed in modern editors and the software techniques of the 1960s can be studied.

Our emulators allow our preserved software to be run on modern hardware, permitting people to experiment with and experience something of 1960s software at first hand — and marvel at the functionality achieved in such a small amount of store.

Marvelling available at: sw.ccs.bcs.org/KDF9/.

Acknowledgements

I enthusiastically acknowledge the contributions made by my co-workers. Without Brian Wichmann's initial inspiration and enthusiasm for resurrecting Whetstone ALGOL, it is unlikely that either of these projects would have started. His subsequent diligence with photography and copy-typing were a cornerstone of the project. Graham Toal and Roderick McLeod similarly kept up the vital flow of copy-typing. Bill Findlay has been most thorough and painstaking in implementing his emulator. Late in the WALGOL project, fingers began to tire and an email plea for more fingers elicited further contributions from Tony Hillman, Peter Stanley, Chris Willis, Roger Broughton and Michael Poole. Special thanks are due to Michael Poole and also to John Thomason who had kept the listings and to Brian Wichmann who had kept his stock of ALGOL test programs.

A WALGOL page's progress to execution

- The page of listing is photographed by BAW to produce ...
- ... a JPEG file, which is emailed to a copy-typist (along with 9 other JPEG files) who produces ...
- ... an initial typescript which is emailed to the collator who parses it.
- Syntax errors are corrected mostly by reference to the photograph.
- Here it meets the typescript of the same page from a different typist.
- Both pages are "canonicalised" and compared using diff -wiB.
- Differences are analysed by reference to the photograph.
- The more accurate version is retained and corrected by reference to the diff output.
- The page is now combined with other pages and assembled, ...
- ... at which point only very few really nasty errors are left (see above).

The RAF Pay and Records Project 1963-4

Michael Knight

The difficulty of shoe-horning large-scale data processing applications into tiny and expensive (by today's standards) computers with inexperienced implementers sometimes makes us wonder how anything ever got done at all. But Michael Knight has been there and returned with tales to tell.

1963 was crunch year for Remington Rand (later Sperry Rand) Univac UK. It had a small customer base of small business systems using the obsolescent, drum memory USS80. It did not have the sales presence to exploit adequately the recently launched plugboard-programmed Univac 1004, last in the line of punched card processors. Univac soon ceded the marketing rights in the UK and in Europe, to ICT, which sold a lot of them. It waited desperately for large system orders, on which decisions, typically, were agonisingly delayed.

At last, like the proverbial omnibuses, three orders came in quick succession: a 1107 for Birmingham Computer Services (BCS, later part of UCC), a 1107 for the RAF and a massive Univac 490-based system for BEA (see *Resurrection 43*).

The RAF Pay and Records system was a very large batch-processing task. Someone in the procurement process had decided it would be interesting to have a large public sector system entirely dependent on paper tape for primary input and output. The 1107 supported respectable paper tape devices, but its software was thoroughly punched card oriented. No punched card peripherals were configured in the RAF system. The central processor featured 32k 36 bit words of main (core) memory with a cycle time of 4 microseconds, which might be halved for memories arranged in two banks. Protection against program corruption was provided in multiples of 2K words. There was also a plethora of registers implemented in an exciting, fast but, as it turned out, transient technology called thin-film. Among these were 16 arithmetic registers, four of which overlapped with four of 15 index registers. In all, this thin-film memory amounted to 128 words, with a cycle time of 0.6 microseconds. The rich instruction set included double-length and floating point arithmetic; but not decimal instructions. The RAF application did not need such riches. It did need fast drum storage for software and other, relatively static, low volume data. It also needed So too fast reel-to-reel magnetic tape drives for file processing. The RAF configuration included FH880 (8 mega-characters)

drums and eight 120 KHz tape drives, both impressive by the contemporary standards of the UK computer industry.

With hindsight, one may wonder why the Univac III was not proposed. This was a big batch processor, somewhat cheaper than the 1107, its tape drives even faster (133KHz). Its instruction set included decimal arithmetic. It was a rather curious design, featuring a 25 bit word. Univac III was never sold in the UK and it may be that Univac UK, confidently anticipating the need to support a community of 1107s and 490s, avoided the additional burden of supporting a third, very dissimilar large system range. Surprisingly, nearly three times as many Univac IIIs were sold as 1107s (96 to 36), although mainly in the USA and only one in Europe (Spain).



Before the coming of discs the author has a quiet smoke

The RAF application was driven by well over 200 different paper forms, completed, often by hand, at stations across the world. Nearly 200 different paper outputs were required, in great volume. The main files reflected a service strength of some 130,000 people. As well as fast paper tape readers, the configuration included three 600 lpm barrel printers. The chosen Executive (operating system) software was the multi-programming EXEC1 (precursor of EXEC8 and not the single application stream EXEC2). The chosen programming language was the macro assembler SLEUTH1, not the even richer SLEUTH2. Why not COBOL? Perhaps we did not trust the efficiency of the compiler. Perhaps it was because the Univac project team lacked COBOL experience; indeed, at the outset some of us also lacked symbolic assembler experience.

The system was to be sited at RAF Innsworth, a non-flying station near Gloucester. Project development was in nearby Air Ministry premises, a grim, single storey, brick-built complex of offices attached to a bleak web of corridors. In summer 1963, a six strong Univac project support team, initially five men and one woman, took up residence in rented

accommodations in Cheltenham. Each member was allowed a weekly return fare to London and £10 per week living allowance. On these now seemingly modest expenses we lived a truly sybaritic existence. Disposal of bottles was a recurrent problem in a non-recycling era.



Each Flowchart should occupy one page!!

Univac had no involvement in the systems analysis task, which was managed by a senior Wing Commander. Its main outputs were flowcharts, one for each input type, specifying the required processing. The charts meticulously followed unnecessarily detailed standards on presentation and symbology as was customary at the time and were published in ponderous A2 sized binders. One standard decreed that each chart should occupy one page, which was fine for most inputs, but not for all. In the latter cases, the pages were extended and gate-

folded as necessary – up to some four metres in some instances. Thus do petty prescriptions collide with common sense.

Programming for these flowcharts was the task of a substantial mixed team of Air Ministry staff and RAF officers, mostly without previous programming experience. The task of the Univac team was to facilitate that work, partly by developing enabling software: to adapt Univac software to accept paper tape input, for example, and to develop standard validation and binary/decimal conversion subroutines. A table-driven approach was developed, defining input and output formats, to facilitate requirement changes and transaction programming. Further Univac tasks were informal mentoring and not least, formal training.

An induction and programming course was devised, which confined itself to the needs of the project. It began, necessarily, with basic concepts, the mysteries of binary and octal and the basics of the RAF configuration. It

excluded, for example, floating point arithmetic and SLEUTH1 macro-instructions and I/O for absent peripheral types. Teaching macro definition seemed an avoidable complication and we feared for program object code sizes if we unleashed the technique. The course also served as an extended aptitude test. Applicants to join the initial core team were mainly civil service clerical and executive officers, few of whom had pre-existing computing knowledge. For some, the main attraction was the prospect of retirement in the vicinity of Gloucester.

Many failed the test, leaving the programming group understaffed. A new initiative was needed. The Univac team staff had been made honorary members of the RAF Innsworth Officers' Mess, a privilege not universally welcomed by serving members. Concerns, we learned at intervals, via Remington House, in Holborn, included lapses of sartorial taste (unmatched luminous socks were instanced), neglected mess bills, indecorous mirth and failure to linger over postprandial coffee in the Mess lounge. Our attendance faded, but not before we noticed the presence of numerous, apparently underemployed young women pilot officers (WPOs). Several of these were seduced onto the programming course and, with some coaching, did relatively well. They joined the programming group.

This imaginative human resource initiative worked well for several months. Eventually, the WPOs became restless, resenting the need to re-visit, when requirements changed routines already completed when they were pining for exotic postings in far pavilions.

"I don't know what's the matter with them" sighed the Ministry chief programmer in a progress meeting.

"I do," growled the grim Wingco i/c systems analysis (subsequently Bursar of Coventry Cathedral). There followed a memorable stream of expletives which your editor forbids me to share with you but the thrust was that Univac had, in certain personnel administration respects, "let us down again".

As an example, my own tasks on the project were:

- devise and participate in the training course
- operate program testing sessions and write a system-specific operators' manual. EXEC1 operating details were scattered through its voluminous programmers' reference manual.
- develop a file information retrieval utility, to search for and extract specified data and print them out from magnetic tape files (SEXPOT)
- develop a print "symbiont" (actually EXEC2 terminology), to drive up to three line printers with spooled files from any combination of drum

or tape drives. The whole program, including buffers and operator dialogue, was required to run within one protected memory segment of 2K words.

Univac mainframes were then still built to order and installation was typically one to two years after order. It was 1965 before the Innsworth system was installed. Early program testing for the RAF was therefore undertaken, always at deeply unsocial hours, at other sites. Two such visits were made to a particle research site outside Paris, where the 1107's day job was to analyse gas chamber images. Two more were made to Regnecentralen, outside Oslo, a university computing facility, followed by several to the bureau system at Birmingham Computing Services. Paris was memorable for Left Bank lunches and nocturnal mud; Oslo for inept efforts at cross country skiing, then debugging on the breezy shore of Oslofjord – not easy with fanfold stationery; Birmingham for generously-pouring waiters at the Albany Hotel and feral nocturnal marauders

This RAF project was a batch processed, administrative computer application typical of its time, except perhaps in scale. Transfer of "other ranks" records was completed in 1968. Oddly, transfer of officers' records was not completed until 1973. In the latter part of 1964 the Royal Army Pay Corps sought tenders for a similar, still larger application, replacing its IBM 1400 equipment. A key requirement was that the new system must be able to read IBM 1400-formatted tape files, to be successfully demonstrated. For Univac I produced this demonstration at BCS. Surprisingly, IBM with their new 360s at that time could not. They won the business anyway.

Subsequently, RAF Pay and Records migrated to an ICL 2900 under VME/B. RAF Innsworth closed in 2008, by which time pay and personnel administration for all three defence services had been consolidated and contracted out to EDS (now HP). The last change was not an unqualified success. Thousands of service personnel were paid the wrong amounts over a long period. Questions were asked in The House. To quote L.P. Hartley "*The past is another country. They do things differently there.*"

Michael Knight was a Univac programmer on the RAF Innsworth project. He can be contacted at michaelknight242@tiscali.co.uk.

An Archive of Early British Computing

Hugh McGregor Ross

The Institution of Engineering and Technology, perhaps better known under its old name the Institution of Electrical Engineers, maintains an archive of books and documents relevant to the interests of its members. It is distinguished in that professional archivists are employed to maintain and support the archive, in particular, to create and update the catalogue of the archive.

Very recently, the website of the Institution has been enhanced and now provides access over the internet to that catalogue.

The Archive is structured as a series of collections. Of course the most famous is that of Faraday. However the archive was generous enough to set up the *McGregor Ross Collection* to accommodate contributions that I was starting to make for the archive derived from my work since the earliest days of electronic digital computers.

This note summarizes the contributions that have been made to that collection. Each one comprises a dossier giving an introduction and a group of items specially written, either by myself or by colleagues, on the specific topic, plus original relevant documents and photographs where these have been available. Identifying and getting these items has, for most dossiers, been a major task.

To access this archive on the internet follow the sequence: Google → theiet.org/archive → search the Archives online catalogue → [Repeat that on its second appearance] → uk0108 naest 176 → Search.

My collection is identified by the code "naest 176". The Search facility is very unforgiving and every search key has to be input exactly.

The catalogue gives the title of each dossier and of each item, sometimes expanded to comprise a brief description. It does not attempt to show in full the items themselves, nor formal abstracts. However, the whole archive is currently housed at the IET headquarters in London, where the items in the archive may be viewed. The archivists are very helpful to enquirers.

The dossiers currently in this collection are summarized below:

Introduction to the Collection (uk0108 naest 176)

[Includes a biographical note, see also Wikipedia and Google.]

Making Computing Standards 1961 to 1989 200 items (uk0108 naest 176D)

Records the setting up of the formal standards-making activity in Britain, with its international equivalents. Offers researchers a facility to investigate the progress of specific standards during those years.

Promotion of Computing Standards 88 items (uk0108 naest 176C)

Includes a complete set of the Informative Listings of Computer Standards published annually in the Computer Users Year Book (naest 176C and naest 176C/3)

Also a complete set of the NCC Guides to Computing Standards (uk0108 naest 176/1/02)

Also a complete set of the bi-monthly NewsLetters *Universe of Characters* published at a critical stage in the evolution of the Universal Character Set standard ISO 10646/Unicode (naest 176/1/05)

Creed Fast Tape Punch and Fast Printer 13 items (naest 176/2)

The punch operated at 300 ch/sec and the dot-matrix printer ran at 100 ch/sec. Includes technical descriptions of both machines.

Ferranti Pegasus computer with punched card machine 1958 to 2001 55 items (naest 176/3)

With contributions, and some photographs, by the engineers who worked on this project, this gives the original story of the Pegasus computer fitted with a special punched card machine for Skandia Insurance, now displayed in the Science Museum.

Digital Electronic Telegraph Switching Systems by STC 55 items (naest 176/4)

The wired-logic STRAD systems and the later computer-based ADX systems, which introduced automatic message switching into the world's telegraph networks.

The paucity of applications for early digital computers 9 items (naest 176B/1 to /9)

A set of documents and diagrams on the theme that digital computers were evolved before the need for their use had been identified and established. A contribution done for the CCS.

Except for the last item, all these original documents exist nowhere else.

It is hoped that this note will encourage other British pioneers of computing to do the same kind of thing.

Postscript Another exceptionally valuable collection of documents on early British computing has very recently been made available in the Archive and its catalogue on the internet. It comprises documents collected by Harry Johnson and a number of contributions specially written by him summarizing his long and distinguished career in Ferranti. It can be found with the search key uk0108 SC MSS 194.

Book Review: Inventing the PC, The MCM/70 Story

Rod Brown

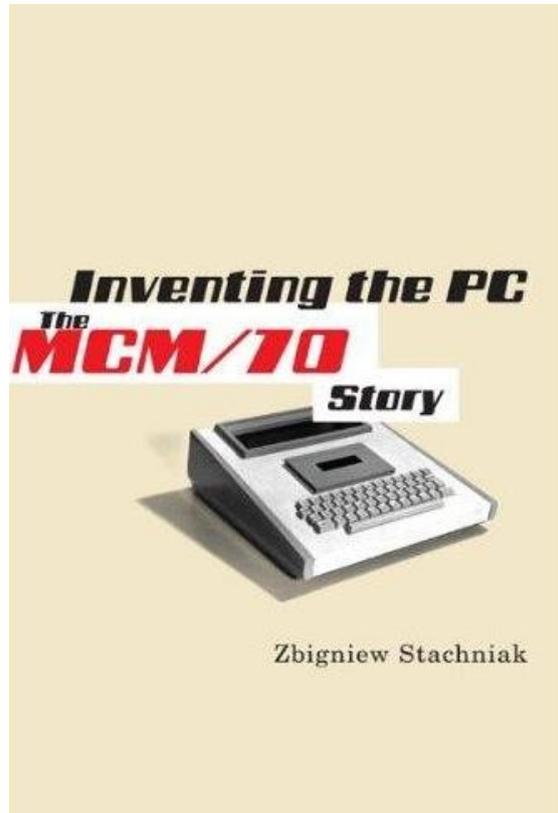
The topic of early Personal Computing is a subject we all think we know all about; until you read this book. This is a very well researched book, both in depth and extent. Investigating the subject, Zbigniew Stachniak has uncovered that little known area between early microprocessor chip designs and the last of the discreet TTL-only small computers, such as the less successful KENBAK-1. At this same point on the time line of computing the PDP machines were hardly even luggable, let alone portable.

The MCM-70 design was ultimately based on the very early 8008 chip, it was also battery powered and portable. The choice of APL as the native language was sensible as it was a mature and well developed language in the era of the early 1970s

However, in a period of rapid early development of the breakthrough required to make a portable computer, with a limited choice of peripheral devices available at the time, the forging of a company and creation of a market had many pitfalls. And it would be the latter rather than the design that decided the fate of this machine.

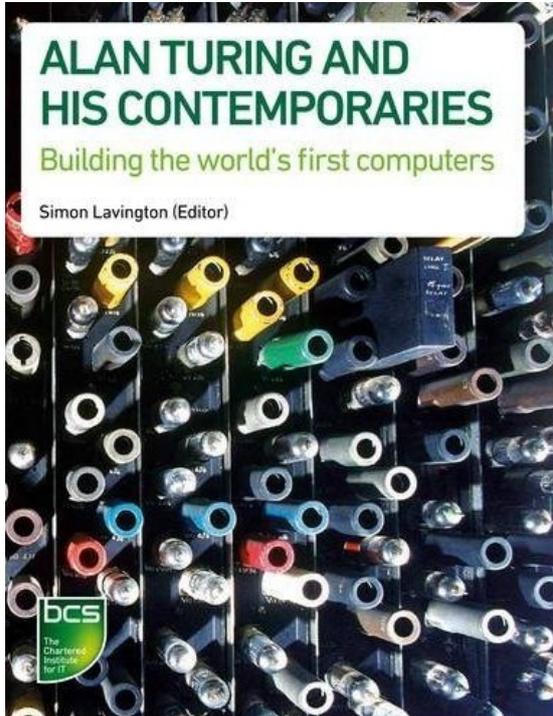
The story of determination to deliver the design and the concept is a testament to the pioneering spirit which formed the foundation of the industry since the early 1950s. Make no mistake, this machine did deliver a working design and several developments spun off as the design matured.

A combination of superb photographs and historic background make this story about a fledgling Canadian company a pleasure to discover. As a story about the birth of the portable personal computer it will inform and delight.



Book Review: Alan Turing and His Contemporaries

Dik Leatherdale



This volume is published in celebration of Turing's centenary. But it is not primarily a book about Turing. Neither is it just about his contemporaries. The subtitle, however, gives the game away – *Building the world's first computers*.

Turing looms large over the first few chapters but does not dominate. We are transported back to a late 1940s world in which if you wanted a computer it was possible, essential even, to build it yourself. From four UK research bodies – Manchester University, Cambridge, the NPL and Birkbeck College emerged four strands of computer

development each of which was taken up by an existing company – Ferranti, Lyons, English Electric and the British Tabulating Machine Company. The fifth, Elliotts emerged without benefit of an academic partner but directly from wartime experience. All five, in time, were destined to become parts of ICL.

Much of the history will be familiar to long-term readers of *Resurrection* but it is fascinating to see them side by side. Other early self-builders such as Imperial College and TRE are not forgotten but their efforts were less influential.

In the final chapter Turing re-appears as his legacy is considered. Why was he for so long a forgotten man and how did his name re-emerge in the 1980s to assume the status of "National Treasure"? Much of the blame for the former lies in his inability to work with others. Much of the credit for the latter belongs in my humble view to his biographer Alan Hodges.

Simon Lavington has brought together four of the Society's most highly regarded authors, each the expert in his particular field. Weaving their stories together, this book presents an evocative perspective on a world whose first-hand memory is now starting to slip beneath the waves.

The Ferranti Mercury at ICI – a Follow-up Note

Tommy Thomas

Having moved from Manchester in 1956 to the Central Instruments Laboratory of ICI it became my task to introduce these new fangled machines to a largely sceptical community who built chemical plant with slide rules very nicely thank you. There were a few pockets of computing activity where scientists relished the challenge of struggling with the machine code of the early Elliott machines.

Throughout the company we soon discovered that individual engineers were amenable to being taught how to program in the Autocode that was first made available to them through the remote use of the Manchester Mark I.

Routines and programs were written in Autocode which, when punched on Creed paper tape, were easily transmitted via the ICI sales telex network to a convenient terminal in Piccadilly, Manchester. These tapes were physically transported down Oxford Road to Owens College and, later in the day output tapes with good error diagnostics would follow the reverse path. Results would be returned via the telex network to the originating engineer anywhere in the ICI worldwide network of factories and design offices.

A key factor in this intriguing early example of remote access computing was the recruitment of Bernard Richards from Tony Brooker's team at the University to staff the Manchester relay station. Bernard subsequently joined the Central Computing Facility of ICI.

We had in effect simulated an early global internet type of operation from 1956 entirely on the strength of the teachability and diagnostic capability of Tony Brooker's early Autocode compilers.

Tony and his colleagues anticipated the need to bridge the gap between the raw coding characteristics of the early machines and the reasonable reluctance of the average user to become subservient to their architecture.

We installed the Mercury computer at Wilton works on Teesside, North Yorkshire in 1958 as a central facility for use by engineers throughout the Company in the UK and overseas.

Management at the Main Board Director level was put under pressure by the then Technical Director (Dr.) Dick Beeching to spend a weekend at

Wilton Works being taught Mercury Autocode. Wilton works was chosen as the site for this central facility because of the proximity of chemical plants and their engineers from a number of ICI Divisions. The sales telex network hub at Wilton was in the room next to the computer room and, through a hatch in the intervening wall, a worldwide computer service was operated.

Users were given 30 minutes of free computer time on the Mercury after which they had to pay by the hour. This strategy was again recommended by Dick Beeching on the basis that if they could not prove the worth of the time used then they should move over to leave room on this limited resource for others more capable.

In 1962, we ordered a KDF9 to replace the Mercury computer. When the KDF9 was delivered in February 1964, the Mercury was being used on a three-shift basis for five or more days a week and the workload was programmed almost entirely in Mercury Autocode.

Our Autocode library contained over 100 programs and several hundred personnel were conversant with the Autocode language and over 2,000 programs were in use. It was the intention at that time to use Autocode on both machines during the transition period and gradually to convert programs and programmers to ALGOL.

An autocode compiler was written for the KDF9 and came into use in May 1964. Another compiler was also provided which contained extra facilities. These were welcomed and suggestions were made for further improvements. ALGOL never took off.

Statisticians and chemical engineers throughout ICI were served well in those years and the digital computer had become an essential tool in the design and operation of chemical plant.

Editor's note: Tommy Thomas has been engaged with computers since 1948 when he took part in developing the early machines at the University of Manchester under Tom Kilburn. From 1966 he was the director of the first of the three Regional Computer Centres at Edinburgh, first proposed in the Flowers Report of that year. Tommy Thomas can be contacted at thomasglengara@mac.com.

North West Group contact details

Chairman Tom Hinchliffe: Tel: 01663 765040.

Email: tom.h@dsl.pipex.com

Secretary Gordon Adshead Tel: 01625 549770.

Email: gordon@adshead.com

David Barron : A Personal Tribute

David Hartley

David Barron, emeritus professor of Computer Science at Southampton University, died on 2nd January 2012 aged 76. It will be for others to produce formal obituaries; this short article is being written as a personal tribute by one who worked with David in the early Cambridge computing days and who is privileged to have been a close personal friend for over 50 years.



I met David when I joined the Cambridge Maths Lab in 1958; I was a post-graduate student, while David had been a user of EDSAC. By the time we met, he had climbed to the dizzy heights of a University Lectureship. Having myself progressed to a research studentship, we found ourselves working closely on the EDSAC 2 computer to which David had made major contributions; to the microcode, to programming techniques, training users and teaching computer science students. We then worked together in pioneering software for the Titan (Atlas 2) computer including the development of CPL.

David left Cambridge in 1967 to take up the founding chair in Computer Science at Southampton and I climbed into his vacated lectureship at Cambridge. We remained in close touch ever since. He made an enormous contribution to developing the Southampton department and became famous for his skills in interpreting the subject. His writing and lecturing abilities were second to none. In recent years he returned to Cambridge to give two seminal lectures – one on the 60th anniversary of the EDSAC, the other in the Cambridge tribute to the late Sir Maurice Wilkes. Although by then in failing health, he showed that he had never lost his touch to inform and entertain.

Our friendship is exemplified in a rather quaint custom we developed. In the early days the Lab was quite small and, since there were three of us called David (Barron, Hartley and Wheeler), the name rather dominated. We avoided confusion by simply referring to one another as David (you), David (me) and David (him). It is sad that there is now only one of us left.

Forthcoming Events

London Seminar Programme

- | | | |
|-------------|--|----------------------------|
| 16 Feb 2012 | Bloodhound on my Trail: Building the Ferranti Argus Process Control Computer | Jonathan Aylen |
| 15 Mar 2012 | Turing and his Contemporaries | Simon Lavington et al. |
| 19 Apr 2012 | Centring the Computer in the Business of Banking: Barclays 1954-1974 | Ian Martin & David Parsons |
| 17 May 2012 | Turing | Brian Carpenter |

London meetings take place in the Fellows' Library of the Science Museum, starting at 14:30. The entrance is in Exhibition Road, next to the exit from the tunnel from South Kensington Station, on the left as you come up the steps. For queries about London meetings please contact Roger Johnson at r.johnson@bcs.org.uk, or by post to Roger at Birkbeck College, Malet Street, London WC1E 7HX.

Manchester Seminar Programme

- | | | | |
|-------------|--------------------------------|-----------|-----------------|
| 21 Feb 2012 | Hartree's Differential Project | Analysers | Charles Lindsey |
| 20 Mar 2012 | EDSAC Replica Project | | Chris Burton |

North West Group meetings take place in the Conference Centre at MOSI – the Museum of Science and Industry in Manchester – usually starting at 17:30; tea is served from 17:00. For queries about Manchester meetings please contact Gordon Adshead at gordon@adshead.com.

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/lecture.htm. Details are also published in the events calendar at www.bcs.org and in the events diary columns of *Computing* and *Computer Weekly*.

Museums

MOSI : Demonstrations of the replica Small-Scale Experimental Machine at the Museum of Science and Industry in Manchester are run each Tuesday between 12:00 and 14:00. Admission is free. See www.mosi.org.uk for more details

Bletchley Park : daily. Exhibition of wartime code-breaking equipment and procedures, including the replica Bombe, plus tours of the wartime buildings. Go to www.bletchleypark.org.uk to check details of times admission charges and special events.

The National Museum of Computing : Thursday and Saturdays from 13:00. Situated within Bletchley Park, the Museum covers the development of computing from the wartime Tunny machine and replica Colossus computer to the present day and from ICL mainframes to hand-held computers. Note that there is a separate admission charge to TNMoC which is either standalone or can be combined with the charge for Bletchley Park. See www.tnmoc.org for more details.

Science Museum :. Pegasus "in steam" days have been suspended for the time being. Please refer to the society website for updates. Admission is free. See www.sciencemuseum.org.uk for more details.

CCS Website Information

The Society has its own website, which is located at ccs.bcs.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, where there is other material for downloading including simulators for historic machines. Please note that the 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.murrell@tnmoc.org, or by post to 25 Comet Close, Ash Vale, Aldershot, Hants GU12 5SG.

Committee of the Society

Chair: **Rachel Burnett FBCS:** rb@burnett.uk.net
Secretary: **Kevin Murrell MBCS:** kevin.murrell@tnmoc.org
Treasurer: **Dan Hayton MBCS:** daniel@newcomen.demon.co.uk
Chairman, North West Group: **Tom Hinchliffe:** tom.h@dsl.pipex.com
Secretary, North West Group: **Gordon Adshead MBCS:** gordon@adshead.com
Editor, Resurrection: **Dik Leatherdale MBCS:** dik@leatherdale.net
Website Editor: **Alan Thomson MBCS:** alan.thomson@bcs.org
Archivist: **Hamish Carmichael FBCS:** hamishc@globalnet.co.uk
Meetings Secretary: **Dr Roger Johnson FBCS:** r.johnson@bcs.org.uk
Digital Archivist: **Prof. Simon Lavington FBCS FIEE CEng:** lavis@essex.ac.uk

Museum Representatives

Science Museum: **Dr Tilly Blyth:** tilly.blyth@nmsi.ac.uk
MOSI: **Catherine Rushmore:** c.rushmore@mosi.org.uk
Bletchley Park Trust: **Kelsey Griffin:** kgriffin@bletchleypark.org.uk
TNMoC: **Andy Clark CEng FIEE FBCS:** andy.clark@tnmoc.org

Project Leaders

Colossus: TBA
Elliott 401 & SSEM: **Chris Burton CEng FIEE FBCS:** cpb@envex.demon.co.uk
Bombe: **John Harper Hon FBCS CEng MIEE:** bombe@jharper.demon.co.uk
Elliott: **Terry Froggatt CEng MBCS:** ccs@tjf.org.uk
Ferranti Pegasus: **Len Hewitt MBCS:** leonard.hewitt@ntlworld.com
Software Conservation: **Dr Dave Holdsworth CEng Hon FBCS:** ecldh@leeds.ac.uk
ICT 1301: **Rod Brown:** sayhi-torod@shedlandz.co.uk
Harwell Dekatron Computer: **Johan Iversen:** jo891979@talktalk.net
Computer Heritage: **Prof. Simon Lavington FBCS FIEE CEng:** lavis@essex.ac.uk
DEC: **Kevin Murrell MBCS:** kevin.murrell@tnmoc.org
Differential Analyser: **Dr Charles Lindsey FBCS:** chl@clerew.man.ac.uk
ICL 2966: **Delwyn Holroyd:** delwyn@dsl.pipex.com
Analytical Engine: **Dr Doron Swade MBE:** doron.swade@blueyonder.co.uk
EDSAC: **Dr Andrew Herbert OBE FEng FBCS:** andrew@herbertfamily.org.uk

Others

Prof. Martin Campbell-Kelly FBCS: m.campbell-kelly@warwick.ac.uk
Peter Holland MBCS: p.holland@talktalk.net
Pete Chilvers: pete@pchilvers.plus.com
Dr David Hartley FBCS CEng: david.hartley@clare.cam.ac.uk

Point of Contact

Readers who have general queries to put to the Society should address them to the Secretary (see page 40 for contact details). 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, however need only notify their change of address to the BCS, separate notification to the CCS being unnecessary.

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 Printed by the British Computer Society

© Computer Conservation Society