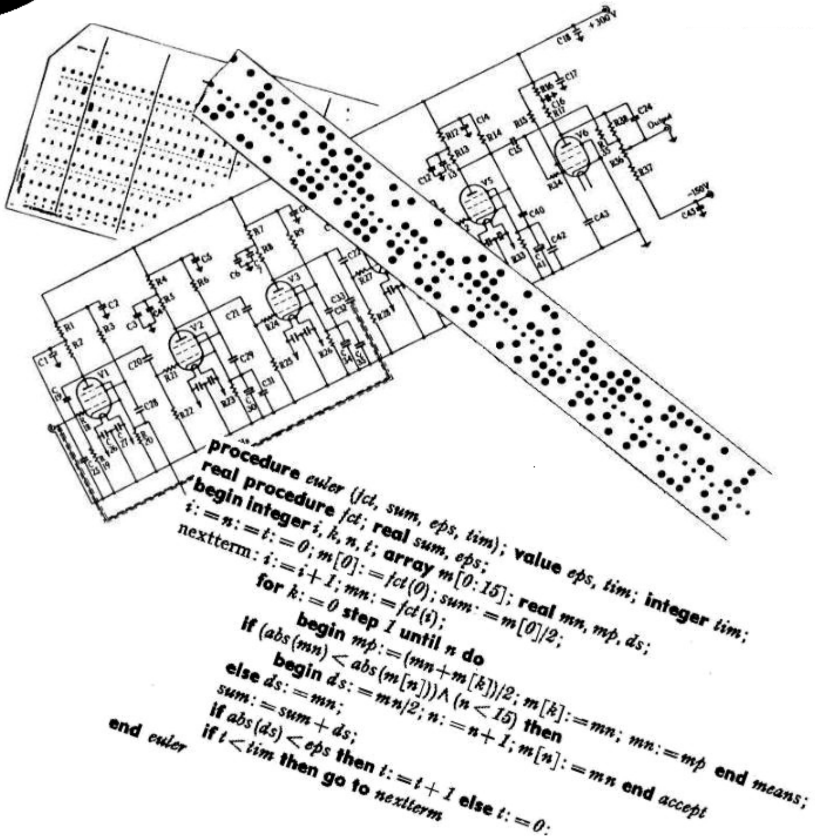
Issue Number 74 Summer 2016

RESURRECTION

The Journal of the Computer Conservation Society



```
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;
        if (abs(mn)<abs(m[n]))∧(n<15) then
            begin ds:=mn/2; n:=n+1; m[n]:=mn end accept
        else ds:=mn;
        sum:=sum+ds;
        if abs(ds)<eps then t:=t+1 else t:=0:
        if t<tim then go to nextterm
end euler
```

Computer Conservation Society Aims and objectives

The Computer Conservation Society (CCS) is a co-operative venture between BCS, The Chartered Institute for IT; the Science Museum of London; and the Museum of Science and Industry (MSI) 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 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 BCS, fees from corporate membership, donations and by the free use of the facilities of our founding 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.

The CCS also enjoys a close relationship with the National Museum of Computing.

Resurrection
The Journal of the Computer Conservation Society

ISSN 0958-7403

Number 74
Summer 2016
Contents

Resurrection Subscriptions

We should remind subscribers that this is the final issue covered by their 2015-6 subscriptions.

You may renew at *www.computerconservationsociety.org/subscribetoresurrection.htm*.

The cost is unchanged at £10 for the next four issues.

BCS members and certain former BCS members continue to receive *Resurrection* free of charge.

Society Activity

EDSAC Replica — *Andrew Herbert*

Inevitably commissioning is showing up a number of systems integration issues lead to delays since resolving such things needs those who understand all the systems being integrated to be present at the same time, and often a rework of one sub-system to be completed before others can move forward. In consequence we have pushed out the date for completion of a fully working replica until autumn 2017.

The focus remains integration of the coincidence system with main control and order decoding so we can demonstrate EDSAC fetching instructions sequentially from store and decoding (but not executing them). Modifying circuits to produce correct logical outputs from signals with gentle rising and trailing edges is a recurrent problem in this work by John Pratt, James Barr and Tom Toth, together with operating within the constraints on how quickly EDSAC flip-flops can be set and reset repeatedly. Useful lessons and general techniques are being learned that should help others when it comes time to add their units to the system.

To facilitate putting chassis in racks upside down for testing and monitoring purposes Alex Passmore and Les Ferguy have slightly modified our power plug and socket system so that an extender cable can be used when a chassis is flipped over without risk of misconnection into the power distribution system.

Peter Linington has delivered a working set of short delay lines in a temporary rack. The aim is that these will be connected in place of the Digital Tank Emulators developed by Nigel Bennée. Peter is now preparing for the manufacture of long delay lines for the main store.

Nigel Bennée continues with constructing and commissioning of the arithmetic system, anticipating delivery to TNMoC when order decoding and main control are sufficiently commissioned.

Chris Burton is designing the second display unit for the operators' desk. The requirement to display the sandwich digit in long words presents the challenge of achieving fly-back in less than a digit time, which is not feasible.

We have begun a conversation about how the operators' desk should be arranged in the replica. Peter Lawrence has reviews the several configurations throughout the life of EDSAC and our inclination is to go with a later design which will be more convenient for giving demonstrations that the 1949 layout.

John Sanderson continues with the design of the electronic circuits for the initial instructions loading unit and engineers' switches.

The tape reader circuit designed by Martin Evans is now under construction by David Milway.

The next anticipated steps are:

1. Commission instruction fetch and decode cycle with coincidence and real delay lines and demonstrate sequential instruction fetch and decode.
2. Move arithmetic system from Cambridge to TNMoC
3. Demonstrate execution of "Load Multiplier" (H) order to verify order decoding, main control and arithmetic sub-system data paths.

Ferranti Argus (Bloodhound) — *Peter Harry*

The restoration of the Argus 700 and the Bloodhound simulator it controls is now complete. The restoration started four years ago when the Bloodhound Missile Preservation Group recovered the Bloodhound simulator, with its Argus 700, from a very damp and non-environmentally controlled place where it had resided for over 20 years. During the four years taken to carry out the restoration the team were involved in acquiring spares, documentation and a conversion of the disc system. The disc conversion being essential to ensure continued running of the Argus due to the obsolescence of the original CDC Wren 1 (FINCH interface) Winchester, now replaced by a solid state SCSI-2 emulation unit.

Activity is now moving from restoration to on-going maintenance and repair of boards. Other work involves emulating the Argus 700's serial bus, the Peribus. The emulation uses TTL to provide the logic and to drive the PeriBus and an Arduino is used to set addressing and data. The objective of the emulation is to enable testing of the system I/O boards without the Argus 700.

Typical of the current maintenance being carried out is the occasional repair of a Farnell switch mode power supply. There are 15 such power supplies used by the Argus 700 and the simulator and the main culprit for the power supply failures are bead tantalum capacitors, now replaced as a matter of course during any power supply repair. Such failures are not surprising due to the prolonged period of time the power supplies have been out of use.

Elliott 803/903 — *Terry Froggatt*

The 803 at TNMoC is running most weekends, without notable issues. The battery seems to be having difficulty holding sufficient charge over the week to work the power on sequence on the following Saturday, so we may need to start thinking about some sort of substitute.

The 903 at TNMoC, which is now in its 50th year, is powered up and demonstrated whenever TNMoC is open to the public or for tours. We've had some trouble with the lamp in one of the tape readers going out momentarily, probably caused by the solder contact on the top of the lamp (which is a car fog lamp bulb) overheating.

Keo Films have been making a documentary on the history of meteorology, and were keen to demonstrate how Edward Lorenz discovered Chaos Theory in the early 1960s on his Royal McBee LGP-30, using a computer of similar power, namely the Elliott 903. So in March I found myself transporting the plotter from my 903 to TNMoC, armed with programs that I'd written to plot Lorenz's two divergent graphs. I was worried that after the plotter had drawn the first graph, it might not find its way back to the same starting point for the second graph, but it was spot on, and everything went reasonably smoothly.

The program was broadcast on BBC4 on Monday June 6th 2016.

Resurrection — *Dik Leatherdale*

Please accept our apologies for the mix-up in the distribution of *Resurrection 73*. A clerical error at the printers resulted in almost everybody acquiring somebody else's forename. Please rest assured that the master membership list remains intact and uncompromised.

ICL 2966 — *Delwyn Holroyd*

Alan is nearing completion of a moving demonstration for one of the EDS200 drives which was partially disassembled some years back to show the inner workings. A motor drives the spindle slowly and a cam system is used to move the head carriage.

The SCP has crashed a few times, bringing the system down, but seems to have stabilised following a few board reseats.

We had a visit from two former operators from Tarmac. Seeing the system in operation brought back many memories for them, and they were able to fill in some blanks in our knowledge of the history of the system. In particular it was confirmed that the system ran batch jobs at night with interactive access during the day.

Some work has been done on bringing the second SCP monitor back to life. The main stumbling block was an open-circuit winding in the transformer, but this has now been swapped with one from a recently donated 7561 with a damaged CRT. Fault-finding on the scan board continues, not helped by the discovery that the last page of the schematic diagram is missing from the aperture cards!

Harwell Dekatron/WITCH — *Delwyn Holroyd*

An audio-visual system has been installed next to the machine. This consists of a large TV on a swivel arm together with video switching to allow input from VGA, HDMI or a media player. A new introductory video is currently in production.

The machine spent a couple of days stopping unexpectedly in the middle of programs, however the problem cleared during fault-finding and hasn't recurred since. It was probably caused by dirt in a relay contact.

A painting of the machine created by artist John Yeadon back in 1983 has been rediscovered on display in a Manchester bar. At the time John made the painting the machine was on display in a non-working state at Birmingham Museum of Science and Industry and looking slightly sorry for itself.

Kaldip Bhamber recently acquired the nearly life size painting to display in her Jam Street Cafe Bar. She had no idea what it portrayed until stumbling upon the TNMOC appeal to find the painting.

A small delegation from TNMOC visited last month to view the painting and meet the artist.

The portrait can be seen at the Jam Street Café Bar, 209 Upper Chorlton Road, Whalley Range, Manchester M16 0BH.

Our Computer Heritage — *Simon Lavington*

Cataloguing

Of the 124 box-files of catalogued OCH-related documents, the final group of 17 was collected by the Science Museum on 28th April. This just leaves a separately-catalogued set of Honeywell manuals, which are due to go to the Jim Austin Computer Collection.

There have been a few minor additions to the catalogue for box MA12 since the March OCH Report, resulting from some Atlas documents that only came to light in mid-March. A new version of this catalogue will be uploaded shortly. The items themselves were included in the batch of 28 box-files collected by the National Archive for the History of Computing on 22nd March.

Updating OCH pages.

This is an on-going process. Reasons for the updates in progress include: changes to external URLs; factual corrections; the insertion of references to the new OCH document-catalogues. At the time of writing (29/4/16) five changed OCH pages (pdf files) await uploading.

Succession-planning.

I will be carrying on with this activity for a few more weeks. In the meantime, we have begun talking to a regular attendee of CCS meetings who may be interested in helping with the OCH project.

CCS Website Information

The Society has its own website, which is located at *www.computerconservationsociety.org.* It contains news items, details of forthcoming events and also electronic copies of all past issues of *Resurrection*, in both HTML and PDF formats, which can be downloaded for printing.

At *www.computerconservationsociety.org/software/software-index.htm* can be found emulators for historic machines together with associated software and related documents all of which may be downloaded. Note that this part of our website is under development. Further material will be added in due course.

Software — *David Holdsworth*

Leo III — Intercode and Master Routine

We have produced an attempt at recollection of the story of the Leo III software resurrection project at: *sw.ccs.bcs.org/leo/BlowByBlow.htm*.

There have been some recent updates to this document, and we have given up hope of getting the programme trials system (PTS) to work. The problem seems to be in the interface between the Master Routine and the Intercode Translator. As reported earlier, the three software products that we have implemented were not exactly compatible versions and we were pleasantly surprised when we got things to work by doing little more than lying about the version number. Our suspicion is that failure of PTS indicates that further incompatibilities remain.

Our search for mag tapes that might contain the CLEO compiler has not added to the list of 7 locations, the most promising of which is the National Museum of Scotland, where they also have Leo III tape decks. I have made initial contacts with the Museum, but a visit has to wait until after their new gallery is operational, autumn at the earliest.

Kidsgrove Algol and KDF9 Usercode

The copy-typing of the code has fizzled out. However, our tests all get through the first two bricks, and grind to a halt trying to enter the missing brick 20. Actually there is one test of invalid Algol 60 which gets correctly reported. It is pretty well certain that there will still be typographical errors, but results are nontheless encouraging. Brian Wichmann is manfully typing KAB01 and has found several typos in comments, and only a single real one of substance. It must be stressed that we do not have copy-typed confirmation of all of the code. We have corrected some typos that have been found rather by chance because the code just looked wrong. Our experiences with Usercode (see below) suggest that we may not need a second copy.

Kalgol compiled Algol 60 programs into Usercode and another missing component of the Kalgol system is the Usercode compiler that then generated the binary program. We do have our modern version **`kal3.c`**, but among the Science Museum's KDF9 possessions has come to light a listing of the paper tape version of the Usercode compiler (KAA01) which Brian Wichmann manfully photographed and copy typed. This can be browsed at *kdf9.settle.dtdns.net/KDF9/kalgol/DavidHo/usercode2.htm*. The lines giving the page numbers are hot links to the photographs.

The file **`little.txt`** is a KDF9 Usercode program that tests that all the mnemonics generate the correct bit patterns. It works when run in load-and-go mode with our resurrected KAA01.

Sadly, one page of the listing did not get photographed, so that code which tries to enter that page fails, so we cannot yet do direct memory access or reference tables.

This has been achieved by an initial copy typing and proof-reading, but then lots of errors have been detected by our kal3.c assembler, and subsequently running the code. The little.txt program pointed us to lots of the errors. The hot links to the images which have been embedded in the listing form **`kal3.c`** make for very quick and effective checking once the general area of an error has been located. Just look in the emulator trace for a jump to the error routine when you get a report that valid Usercode is wrong.

Good assembler syntax checking and good emulator diagnostics maybe able to take the place of a second copy-typed copy. In the case of Kalgol, we are working with OCR output that has been proof read. Even if it is not quicker, it is a lot more fun.

Our readme file has some more detail and links to various web pages: *kdf9.settle.dtdns.net/KDF9/kalgol/DavidHo/readme.htm*.

We believe that our chances of recreating brick 20 will be much improved when we have digested the documentation that Brian Wichmann and David Huxtable have previously seen in the Science Museum archives, to which we now have some access. The cataloguing of the material is not yet complete, and we need to spend more time in Blythe House. I am contemplating coming down to London specially to do this. Unfortunately David Huxtable is not currently able to participate.

There is previous evidence of institutional loss of computer printout that was deposited for safe keeping. It seems that museums and libraries are conditioned to see themselves as preserving physical artefacts. The abstract nature of software would appear to disqualify it as an artefact of historical significance. I reckon that Kidsgrove Algol is the best piece of 1960s software that we have so far encountered, and we have demonstrated our ability to make the first two passes work.

There is much useful information in the four volumes of the KDF9 Service Routine Library Manual. Despite several feelers, I have so far failed to locate another copy. I will digitise it somehow, but the speed and convenience of a sheet-feed scanner carries a risk of damage, a risk I am loth to take if my copy appears to be unique.

Analytical Engine — *Doron Swade*

We have been pecking away at Babbage's original design drawings for some while now and have found with regret that we are unable to reverse engineer a coherent and consistent understanding of the Analytical Engine from the mechanical drawings alone. There are some 300 drawings and some 2700 Notations — descriptions of the mechanisms using Babbage's language of signs and symbols. There were three phases of design — early, middle, and late. There is overlap between these, ad hoc upgrades, fragmentary and sparse explanation, where there is explanation at all.

It remains unclear whether any of these three phases is graced by a complete design. This in itself would be unfortunate but not catastrophic as omissions of mechanisms can be devised provided their intended function is fully understood. The immediate problem is that the extent of incompleteness is not clear. The work of the late Allan Bromley in decoding the AE designs in invaluable but he published only a small part of his substantial findings and these are based on only part of the archive. While much is understood about many of the main mechanisms and the general scheme, there remain fundamental aspects control and sequencing that are not yet well understood and have resisted further illumination.

To achieve a more comprehensive understanding of the designs Tim Robinson in the US is going through the entire Babbage archive (some 7000 manuscript sheets) and producing a cross-referenced searchable data base. The purpose of this is to marshal all known sources so that we have a bounded idea of all relevant material. The intention is to reveal any explanations and/or drawings that Babbage might have left that have not yet come to light.

A second line of attack is to the 2,700 Notations for the AE using the newly acquired knowledge of the Mechanical Notation. Allan Bromley maintained that the Notations were indispensible to his understanding. However he did not publish how he had used the Notations and he was the last to use the Notations as an interpretative tool. The hope here is that the Notations will provide some of the missing information about logical control and moreover give insights into design strategy. In parallel with Tim's comprehensive data base index I am going through the 20 volumes of Babbage's 'Scribbling Books' identifying all material on the Mechanical Notation — a fast-track way of accessing this specific material.

Currently the stages envisaged for the project are:

1. Finalising research of the original design
2. Specifying a viable version
3. Computer modelling and simulation/3D printing
4. Manufacture and construction

We need more hands to the pumps and have latterly diverted some effort to fundraising. We have a three-year plan at the end of which we expect have the requisite understanding of the designs, a platform from which to specify a viable version of an AE that is historically authentic, and to have trialled tools for modelling and simulation. The funding proposal includes pulling in requisite expertise including modellers and mechanical engineers.

In summary, we have had to bite the bullet with the realisation that without a concerted assault on the sources, a fuller understanding of the Engine design will not be forthcoming. We need to understand the intentions of the design well enough to identify missing mechanisms and understand their intended purpose well enough to devise fill-ins that are consistent with Babbage's design style. Effort is now divided between continued study of the designs and fundraising to ramp up the effort.

Manchester Baby (SSEM) — *Chris Burton*

Nothing outstanding to report as the replica continues to provide a good service of demonstrations to the public on three or four days per week. On a typical day the volunteer team might talk to 20-60 people plus school or college groups. It has been noticeable that over the last six months or so, a few visitors have engaged in very detailed discussions with the volunteers about the detailed working of the machine, particularly the mechanism of storage. This is a gratifying justification for the existence of the replica, which prompts questions that might not otherwise have been thought of. The members of the volunteer team are to be commended for their continued enthusiasm for engaging with visitors.

The machine itself continues to perform well, usually using the Control and Accumulator CRTs and semiconductor store backup. Occasional mysterious faults occur — for example the little DC power unit to supply the heaters of the Y-scan summing diodes became intermittent. It took a couple of days to conclude that a brass stud type of rheostat control had got dislodged.

IBM Group — *Peter Short*

Electromatic Typewriter

Our Electromatic typewriter has arrived. On inspection we believe this model may have been built between 1933, when Electromatic was acquired by IBM and 1935, when IBM's redesigned version was released. Investigations continue.

3179

We have also now obtained the 3179 colour display terminal mentioned in the previous report. Unfortunately it was damaged in transit, showing signs of being dropped from a great height. The monitor case is cracked in a number of places, but the machine appears to still be functional. This is a Model 2, which was specifically designed for connection to AS/400 and related mid-range systems.

The only positive is that we obtained a full refund from eBay for the shipping costs plus half the cost of the machine.

CashPoint Model

The model of the 2984 Lloyds CashPoint continues slowly. We have a full set of keytops printed on a 3D printer and finished off with vinyl transfers, and the keyboard & dispensing enclosure is partly built.

Loans to Other Organisations

A lesson in managing loans to other organisations. Some years ago the museum loaned a number of artefacts to an organisation in Winchester aimed at teaching children about technology. This was somewhat before my time, and the loan agreement was not followed up annually as it should have been, until we discovered it in the files last year. We requested return of our artefacts after we learned that two items had gone missing.

In spite of a comprehensive search the missing items have not been located and we must assume lost. We are always willing to lend out items but in future we will need to inspect the security measures at the display location and carry out regular checks.

Out and About — *Dik Leatherdale*

The period since *Resurrection 73* went to press has been particularly busy with invitations to visit people and places far and wide.

Rutherford Appleton Laboratory

On page 15 you will find a report of a visit to RAL in the company of Tony Pritchett — UK pioneer of computer generated films.

Unveiling of HEC1 at TNMoC

Next, off to TNMoC for the unveiling of the prototype of the British Tabulating Machine Company's Hollerith Electronic Computer (a.k.a. the ICT 1200 Series). Dr "Dickie" Bird the designer was on hand for the ceremony at which CCS stalwarts Kevin Murrell, Roger Johnson and Delwyn Holroyd all spoke. Dr Bird despite his advancing years (he is 90) remains a witty and robust speaker with an engaging twinkle in his eye. The CCS donations fund made a major financial contribution to make the new display possible.

LEO Society

Several of us were generously invited to the bi-annual LEO Society reunion at the Inns of Court (no less) in London. I am happy to report that the Society is in rude health and has recently teamed up with Middlesex University for a PhD study of the LEO company and its products funded by a studentship named in honour of the late David Caminer. More at *tinyurl.com/j3oh6ow*.

Deutsches Technikmuseum Berlin

Some 23 intrepid CCS members travelled to Berlin in April to visit the Technical Museum in the company of Professor Horst Zuse, the son of Konrad Zuse and a distinguished computer scientist in his own right. Konrad Zuse was the builder of several very early mechanical and electro-mechanical automatic calculators — arguably the first computers according to what definition you favour. A fine collection of original and replica Zuse computers was on display. Prof. Zuse gave us a very interesting talk and a demonstration of his re-creation of the relay-based Zuse Z3.

Horst Zuse and his Zuse Z3 replica

News Round-Up

LEO/EE/ICL hands who knew Hartree House (LON23), the office above the Whiteleys department store in London's Bayswater may be interested to hear that there has been a recent a proposal to demolish the building.

It has to be said that Hartree House was not exactly a prestige office, but after ICL jumped ship a comprehensive refurbishment of the building incorporating the listed façade and the famous staircase was carried out opening in 1989. Less than 30 years on, all this is now apparently considered passé. More details at *tinyurl.com/j2az3pg*.

101010101

We regret to report the passing of Andy Grove who in 1968 quit Fairchild Semiconductor to join former colleagues Robert Noyce and Gordon Moore at the fledgling Intel. Assuming the role of chief executive in 1987, he saved Intel from oblivion by withdrawing from memory chip production and entering the world of microprocessors the field in which Intel remains dominant to this day. The New York Times obituary at *tinyurl.com/hgwlm3e* is recommended.

101010101

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.

Members who move house or change email address should notify Membership Secretary Dave Goodwin (dave.goodwin@gmail.com) of their new address. Those who are also members of BCS, however, need only notify their change of address to BCS, separate notification to the CCS being unnecessary.

Queries about all other CCS matters should be addressed to the Secretary, Roger Johnson at *r.johnson@bcs.org.uk*, or by post to 9 Chipstead Park Close, Sevenoaks, TN13 2SJ.

Can you Help?

David Holdsworth writes —

"I have copies of the four volumes of the KDF9 Service Routine Library Manual which I would like to scan, but do not want to risk putting the pages through a sheet-feed scanner unless I can identify a copy from which I could copy any pages that get wrecked by the scanner.

I would like to make contact with anyone who also has copies of any or all of these four volumes."

If anybody can help, David can be contacted at *ecldh@leeds.ac.uk.*

Javier Albinarrate heads a project at the Museo de Informática, Buenos Aires, to build a full-scale (but non-working) replica of the Ferranti Mercury computer that was delivered to Buenos Aires University in September 1960. The project is making good progress (see picture). Lots of help has been given by NMS Edinburgh who, as you know, has the remains of a more-or-less complete Mercury.

The Museo de Informática is anxious to acquire some authentic peripherals, for which they hope to get help with shipping costs. In particular, they are looking for the following:

Ferranti TR5 paper tape reader (300 ch/sec.)
Creed 25 Tape puncher
Creed 75 Teleprinter
Creed 6S5 Tape reader.

If anyone knows of any of the above devices that might be donated, or long-term loaned, to the Museo then Javier *(javier@albinarrate.com)* would be eternally grateful!

Animating Centipedes, *Alien* and *Channel 4*

Victoria Marshall

> In the mid-1960s, a young electrical engineer was working at the BBC making educational programmes. Influenced by the computer animations being created at Bell Labs in America, he took a job at the Institute of Computer Science where he had access to London University's Ferranti Atlas 1 computer, the "middle brother" of the Manchester and Chilton Atlas 1 computers. Six months later he had created a two-minute animation of a fantastical centipede having a spot of bother with a mischievous segment.

The animator's name was Tony Pritchett, and the film was *The Flexipede*, the first computer-generated character animated film in the world, with a sound track of "found sounds" including a creaky office chair, a Jew's Harp, a garage hoist, and Tony himself, gulping. The film was first shown publicly in London in 1968 at *Cybernetic Serendipity*, the world's first international exhibition of all aspects of computer-aided creative activity.

A few years later, he became a Research Fellow at the Open University in Milton Keynes "advancing the technology of computer animation for the benefit of the OU." It did not advance very far however once they realised the upgrade costs involved [1] so facilities at the Atlas Computer Laboratory (now part of STFC's Rutherford Appleton Laboratory) were used instead. The OU was broadcasting courses through the BBC, so Tony's job was to create animated diagrams and simulations for their science and maths courses.

In 1979, the film *Alien* was released and became one of the world's top-grossing films. But Ridley Scott is a very exacting person to work for and it took several attempts by Tony, Brian Wyvill and several others to finally come up with "Some computer output which makes the status of the computer orbit logistics clear to the viewer. Savvy?" It is a bit blink-and-you-miss it, but if you watch the on-board monitors of the Nostromo as it comes into orbit around the planet you can just about spot the results of their work. Tony's docking sequence also cropped-up in *Blade Runner* a few years later.

Tony's other contribution to *Alien* was the simulation of a moving x-ray of John Hurt's insides. Unfortunately, that ended-up on the cutting room floor, but not before he had met an up-and-coming young actress who wafted into Ridley Scott's offices one day: "Is there a laundromat round

[1] At the time they had a Hewlett-Packard computer running the student computer service via multiple teletypes on phone lines to the study centres; the only programming language available was BASIC.

here? My name is Sigourney." "What sort of name is Sigourney?" he asked in puzzlement. "What sort of question is that?" he was asked in return. [2] [3]

Channel 4 was launched in 1982. This was supposed to be a new concept in television broadcasting, a radical departure from the *BBC*, requiring a new and radical identity. Tony animated the logo using John Lansdown's Tektronix 4014, a BASIC-only graphics computer, to output vector forms onto a plotter; these were then hand-coloured. Unfortunately the result was rejected by Martin Lambie-Nairn as insufficiently high-tech so Tony frantically rang round his contacts at SIGGRAPH in Boston to try and find someone who could help. Triple I (Information International Inc.) in Los Angeles had just finished work on *Tron* so were happy to help, which is why Tony ended-up using a hired Tektronix to squirt 3D vector coordinates down a serial line into their Foonley supercomputer. A mouse driving an elephant! The logo was rendered just in time for the launch of *Channel 4*, and Tony flew back to the UK with less than a week to spare.

Over 40 years later in February 2016 he came back to visit, accompanied by film-maker Kate Sullivan – *www.katesullivan.co.uk* – who is making a documentary about his life and work, provisionally entitled *Tony Pritchett: Cybernetic Serendipity*. Tagging along (his words!) was *Resurrection* editor, Dik Leatherdale, a colleague from the London Atlas 1 days with whom Tony once shared an office. They were shown round by Prof Bob Hopgood, OBE, one of the original Atlas Laboratory staff (now retired) and Victoria Marshall (still employed). A fair amount of imagination was required because while some parts of the building are just the same, others have changed significantly since its construction in 1964. Bob had brought with him photographs of the interior as it had been, so there was much arm-waving and head-scratching while everyone tried to work out what had been where and where we were now.

Part of the European Space Agency now occupies the south side of building, which explains why the surface of Mars and a moon buggy are outside what used to be the Atlas Laboratory main entrance.

The goldfish bowl "Think Room" on the first floor above the original Reception area is still there; so too is the haunted corridor outside what used to be the pigeonholes from the Operations Room.

The further you go towards the back of the building, the less it has been refurbished so Nigel Kellaway (cameraman for the day) wandered around photographing "texture" as he put it… 1960s concrete slab paving, Civil Service door handles, and spindles from the semi-spiral staircase in the

[2] Sigourney is a character from *The Great Gatsby* and the actress was of course Sigourney Weaver.

[3] No laundromats were found, so they went for a drink in a Pinewood bar instead.

original Reception area up which the FR80 colour film processor was man-handled. [4] [5]

Over lunch there was some discussion about how The Flexipede was produced. Tony recalls that he wrote it in FORTRAN IV on the London Atlas1 computer, and used the GHOST package to transfer the output to tape. He then took the tapes to Culham Laboratory to film it using their Benson-Lehner plotter and microfilm recorder.

Tony Pritchett and the Atlas 1 Engineers' Console

The surprise of the day was undoubtedly being asked "Would you like to meet Atlas?" Thanks to some nifty work by one of Rutherford's electrical engineers, the 50-year-old Chilton Atlas 1 Engineers' console was once again brought to life for the occasion. Tony explained that, as a user, he would not normally have been allowed into the machine room so was intrigued to see it.

Rutherford was honoured to welcome such alumni; they in turn were overwhelmed by the science going on here, and astonished at the wealth of computing history we have managed to retain. There is something wonderful about "living history," both for the people who were there and for those who feel as if they were there, and we very much look forward to Kate's documentary.

Victoria Marshall's interest in the history of computing at the Rutherford Appleton Lab was augmented with contributions from Bob Hopgood and Tony Pritchett. Victoria can be contacted at victoria.marshall@stfc.ac.uk*.*

[4] The FR80 microfilm recorder delivered in 1975 only just fitted into the lift to take it up to the first floor. The colour film processor which came with it was much larger however, so had to be man-handled up the stairs.

[5] The FR80 may have been one of the most sophisticated recorders in the world, but until a new calorifier was installed, one wing of the Atlas Laboratory had no hot water whenever film processing was going on.

Computerising in the 1980s

Knut Jürgensen

Being the surprising story of a financial institution which came late, very late, to the world of information technology.

By Sexecutioner - Zoo Köln (Cologne, Germany), January 2010

Have you ever wondered what it must be like to be able to look forwards and backwards at the same time? What a help this would have been if I'd had this type of vision 30 years ago.

A chameleon's eyes move almost like they're mounted on turrets — separate turrets, since its eyes move independently of each other. This allows a chameleon to watch an approaching object while simultaneously observing the rest of its environment. They have a distinctive visual system that enables them to see their environment in almost 360°.

They don't even have to move their heads in order to keep one or even both eyes, on their objective. Some decisions would be infinitely simpler if we could do the same.

Master Data and Master Data Management (MDM) have been around for a long time. It is the way we record, use and store Master Data that has changed so radically over the years. Change in a way which we could not have foreseen.

The goal of this article is to give some of the younger generation an idea of what life was like back then. A life without little computers that fit into your pocket, take photographs, play sophisticated games as well as your favourite music, send messages, and oh yes, make the odd phone call. Computers in those days were beasts. Even a mini-computer required its own air conditioned room and dust free flooring.

In this article, I would like to look back in time to the mid-1980s, when I first started my journey through the fascinating world of computerised MDM. I started my employment in August, 1981, as the accountant and subsequently, the General Secretary of the Oranje Benefit Society (OBS), in Cape Town, South Africa. We had out of date technology with slow response times due to a cumbersome system and we knew we had to change.

We were aware of the fact that we blended in with the other societies, but also that we could become outstanding and out-colour the rest. So, it was at this time, that I chose to computerise a manual data system for my employer, a non-profit seeking Friendly Society. Wikipedia defines a friendly society as follows: - A friendly society (sometimes called a mutual society, benevolent society, fraternal organisation or ROSCA) is a mutual association for the purposes of insurance, pensions, savings or cooperative banking. It is a mutual organisation or benefit society composed of a body of people who join together for a common financial or social purpose. As you can see, friendly society, not because we were a friendly bunch of people, but because of the supportive service provided.

The OBS camouflage however, had to go. Rather like the common misconception that a chameleon changes colour to camouflage itself. This is not the case. In fact, chameleons show their mood or their intention and adapt to temperature changes by displaying various colours. The mood at the OBS was subdued, lethargic. It was high time for progress and to take a bold step forward.

The starting point can be summarised by the following —

1. Total assets of just over ZAR 10 million
2. Number of members about 17,000
3. Technology – one NCR Class 32 Accounting Machine and 2 or 3 electric typewriters
4. A large number of 4 drawer filing cabinets to accommodate members' record cards
5. A few running meters of metal shelving for the remaining documentation pertaining to the members.

The master data database of the members was represented by thousands of membership cards, roughly 10″ × 15″, stored numerically according to membership number, in numerous, good old-fashioned, 4-drawer steel filing cabinets (right).

The cards were pre-printed with areas for hand written member's details and a separate area for membership class and contribution details, which were also hand written. The bulk of the member's card was for machine printed claims details.

This brings me to the aforementioned technology.

NCR class 32 (photo courtesy JFHP)

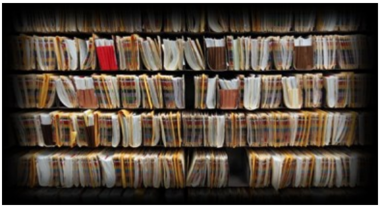
The machine used was an NCR Class 32 accounting machine, manufactured by the National Cash Register Company, later renamed to NCR Corporation. My research indicates that this machine was first launched in 1928 and the version we had was still going strong in 1981. I cannot be sure, but I believe it belonged to one of our predecessor societies, the Cape United Sick Fund, which amalgamated with two other societies in 1968, thus forming the new Oranje Benefit Society.

Let me point out that some of the images might not be exact photographs of the machine we used but visually represent that which I remember. Bear in mind, that in those days we did not run around with our little smartphones taking pictures of everything. Photography was more advanced than the computer but the attitude towards taking pictures of everything that moves had not yet developed into today's standards — that however is another story.

The NCR Class 32 used various program bars and depending on how they were configured controlled the various functions of the machine.

The way it worked was to insert the member's card on the left hand side and a second card on the right. This second card represented part of the Cash Book and was used for accounting purposes. Behind the two cards was a roll of carbonised paper. This carbonised roll was the claims report. The cards and the report formed the bulk of the Master Data.

The balance of the Master Data, application forms, claim forms and all correspondence were kept in individual folders and filed again in numerical order, on shelving. (See below). It was all rather well organised.



In the early years of my employment at the Oranje Benefit Society my first step was to consolidate the various bits of information pertaining to each member. The member record cards were removed from the filing cabinets and combined with the existing folder containing the correspondence. The brown folder needed to take on a new hue — chameleon qualities — be adaptable, flexible, provide readable data and information and turn it into a spectacular, space saving dimension. We needed to show our true colours and my feelings regarding the new computer trends were instinctive. I knew that computers were powerful tools and I was ready for the exciting road ahead.

During my tenure at the Oranje Benefit Society, the number of members varied between 17 and 13 thousand. It was quite a formidable task to update each card manually on a monthly basis with regard to contribution payments. The contribution information was provided by the South African Transport Services (Transnet) in printed form, on continuous stationery. Added to this, benefit claims had to be processed as well. The three dedicated ladies who were responsible for this function had their hands full.

Within the first two years of my employment at the Society the next step in Master Data Management was obvious. We were still in the early 1980s but it was clear to me that computerisation was the next logical step. At this point I could only see what was behind us but was trying very hard to also look forward and visualise the future.

For our purposes, there were basically three aspects which required consideration in order to computerise. Firstly, the price (isn’t it always?), secondly the ease of programming and finally (and this was vital) that Transnet could provide the Society with the members’ contributions in digital form.

It was up to me to convince the Executive Committee, the governing body of the Society, that computerisation was the only option we had in moving forward, in order to provide the service our members deserved.

Being a novice with regard to computers, choosing the most suitable hardware and software supplier was a lengthy and demanding exercise. At the time, all the usual suspects were in the running — IBM, NCR (obviously, after all their many years of loyal service), Olivetti (they did not only produce typewriters) and Hewlett-Packard. After detailed and extensive research I finally decided on the German company Nixdorf Computer AG and one of their model 8870 M-series mini-computers. Selling points were low maintenance and ease of programming, even though it required its own air-conditioned room. My rotating chameleon eye saw that I would have to sell my concept to the Executive Committee, who were the equivalent of a Board of Directors. Armed with all the facts and quotations, I managed to accomplish this, as they were also very keen on moving the Society into the computer age.

Dual Chassis Nixdorf 8870 Model 55

Our configuration was made up of two units similar to the picture (left). If my memory serves me correctly, it was a Nixdorf 8870 Model 55. Unfortunately, there are no images on the internet which show our particular configuration.

The system was supplied as a minimum of two identical chassis (at least from the outside).

The top tier on each chassis held a Storage Module Drive (SMD) and all the required electronics and hardware for the drive. The lower tier held the system backplane and power supply.

Users accessed the SMD drive by lifting an outer protective lid. Underneath there was a clear Perspex lid that helped form an airtight seal for the drive. Opening this Perspex lid would automatically cut power to the spindle motor. When it stopped spinning you could then drop in the drive pack cover and unscrew the pack and then clip on the black case base.

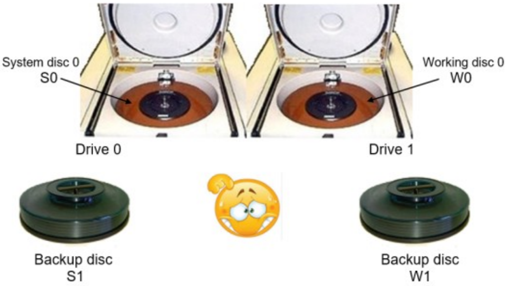
The system used monstrous 14in 66MB removable disc packs for storage and backup (see right). These were manufactured by Control Data Corporation (CDC). Each pack had five discs (three for recording, two for protection) and there were six usable surfaces (five read/write, one servo). We had six disc packs — two per day for three days' worth of backup. The backup procedure was very complicated and fraught with the danger of mixing up the packs, which would result in the loss of critical data. The "End of Day" backup process instructions were provided on screen indicating which disc pack needed to be mounted on which drive, and in which sequence.

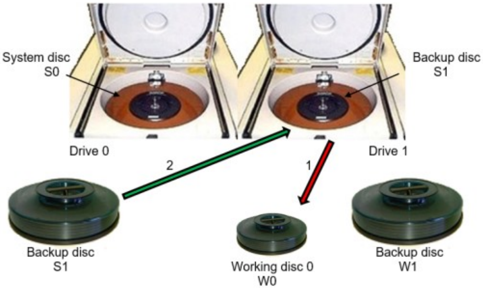
You had to keep your wits about you. One obviously tried to label the cover and disc pack but at the beginning we had a few mishaps with the wrong disc pack being used, as covers were inadvertently switched. This required a fair amount of backtracking in order to recreate the disc packs and then start all over again.

The system operated with two disc packs installed, one referenced S0 (systems disc 0) in drive 0 and one referenced W0 (working disc 0) in drive 1.

Step 1

You had to (1) remove W0 and (2) replace it with S1 (the backup disc to S0). The system would then copy from S0 to S1.



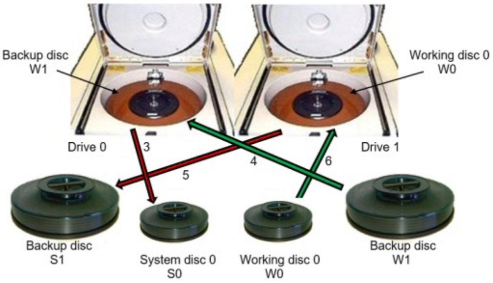


Step 2

Then one would (3) remove S0 in drive 0 and (4) replace it with W1 (the backup disc to W0) and W0 would be (6) put back in drive 1, after (5) removing S1. The system then copied from W0 to W1.

Step 3

At the end, one would replace the W1 pack with the original S0 pack in drive 0.



This involved a fair amount of starting and stopping drives whilst switching disc packs. When the drive was powered up you could hear a vacuum being formed and then there was usually a “whoosh” as the drive heads jumped forward, flying a fraction above each platter on the pack.

The operating system was initially Nixdorf Interactive Real-time Operating System (NIROS) and the customer written software was programmed in Business Basic.

The final piece of the puzzle was that Transnet was able to provide the necessary contributions data on 8in floppy discs, in order to update the members’ contributions. This was later upgraded to receiving the data electronically via a Banksia “Bit Blitzer” modem.

Thankfully, as all our members were employed by Transnet, we were able to call on them to provide us with the bulk of the personal information (name, address, date of birth and so on) in digital form, which was then fed into our new system. Unfortunately, the information which was specific to the Society had to be captured manually (membership class, contribution rate and benefit claims which had an influence on any annual limits).

The software for the Society was entirely customer written as no packages existed at the time. Nixdorf did a great job on the software. It even allowed me, a total novice, to access the programs and to capture changes. This attests to the user friendliness of the software. Initially, the changes were just simple routine amendments, such as when benefits were improved. Later on, the changes included new contribution classes and additional benefits. All done with ease.

The decision to acquire a computer system was finally communicated to the members of the Society in June 1985 via a paragraph in the 1984 Annual Report.

The installation of the computer system was completed in early 1986 and was fully operational by the end of that year.

To sum up, we found that time was saved, we had improved productivity and provided a better quality of service. This was proof of what was at the time a very innovative and bold step. From here on in, both eyes were securely focused on the future.

The Society’s Master Data was now available in an instant, at any time, day or night. I am not aware of any other friendly society computerising its member’s data base at that time.

Although the membership did at one point dwindle to about 13,000 members, the assets grew to R42.2 million by the time I left. This was another one of my major achievements.

Now, after 22 years of additional experience, I am able to see with hindsight, that our camouflaged position of blending in with other friendly societies had indeed moved forward. We had changed the face of the OBS by moving with technological changes. We had taken on the new dimension of digitalisation and my vision had been transported into the reality of the new Society — we were no longer hidden in the friendly society crowd but stood apart.

Chameleon by Ttalk-Kevin Wu on Flickr

Acknowledgement

My grateful thanks to Joe Farr for his invaluable support regarding the technical information of the Nixdorf 8870 M55 computer. His internet pages on "Adventures with the Nixdorf 8870 Mini-Computer" and his e-mails were incredibly helpful.

Knut Jürgensen is a Master Data Administrator currently with an engineering company, based in Germany. Previously he was the General Secretary of the Oranje Benefit Society in South Africa. He can be contacted at knutj@t-online.de.

North West Group contact details

Chairman Tom Hinchliffe: Tel: 01663 765040.
Email: *tah25@btinternet.com*
Secretary Gordon Adshead Tel: 01625 549770.
Email: *gordon@adshead.com*

Birth & Death of the Basic Language Machine

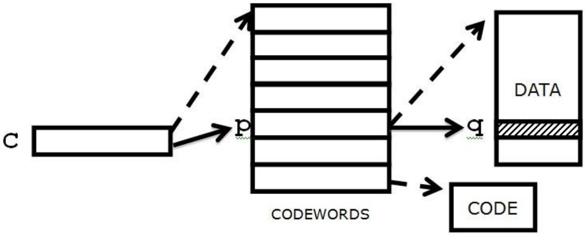
John Iliffe

The Basic Language Machine, constructed in the ICL research department in 1963-68, was the first general-purpose system to break completely with the concept of a linear address space, thereby offering automatic storage management and precise security environments. Although in existence for a short span of time it was preceded by a lengthy gestation period, in which the essential features were demonstrated, and a long "lying-in-state" in which it was further refined.

Rice University

In 1958 I was invited to join the Rice University Project with the aim of developing an operating system and programming languages for a new machine (R1) being built on campus for scientific calculation. Until then I had been managing the IBM 650 and 704 service bureau in London, and welcomed the opportunity to try out new ideas on a machine powerful enough to put them to practical use.

Arriving in Houston in September, I found myself discussing the order code with Professors Graham, Kilpatrick, Salsburg, from Rice, and Bob Barton, from Shell Development. At that 1958 meeting a new method of addressing store was outlined by John Kilpatrick, essentially a generalisation of indirect addressing, which was currently featured in the IBM 709.



THE STANDARD MODEL

A datum (at q) is selected by addressing the codeword C, then stepping through the tree modifying the starting addresses found in successive codewords. In R1 the indices (p,q) were pre-set in registers B1, B2, and the addressing rule automatically followed the path to q. Other implementations supplied indices "one step at a time".

The mechanism described by Kilpatrick might have been used in many ways, but it was seized upon as a means of representing commonly occurring data structures and gaining access to them efficiently. Moreover, if properly controlled the structures could be created, reshaped, or destroyed dynamically under program control. The key to this was a "mapping" element that allowed the user to access data without direct knowledge of memory location, as illustrated in the Standard Model above.

Barton and I came away with that idea, which evolved into the Descriptor used in the Burroughs machines, and the Codeword used on R1. From that point our approaches to design diverged: Barton elected to make the machine fit the language (a variant of Algol on Burroughs B5000), whereas R1 defined the "memory", then looked for ways of exploiting it in system design. Thereafter we had little contact, for we were heading in different directions.

There was no pressure on R1 to provide a variant of some "standard" language or operating regime. The natural tendency of storage management in R1 was to deal with procedures and data as objects that could be combined dynamically to form executable programs. The algebraic coding language (GENIE) exploited these ideas. The entire store was presented as a tree structure (of any required depth) rooted in fixed positions in memory, an individual program being attached as blocks of codewords, data and instructions. A special case of "tree structure" is of course one block of data, corresponding to a conventional store: structure is only elaborated when the problem calls for it.

The Basic Language Project

In 1961 I joined the Ferranti computer department, where the Orion and Atlas were on the point of working. Consideration was being given to a successor (which became the ICL 1900 series), but I was invited (by Stan Gill) to consider a radical redesign based on R1, with which he was familiar. By this time the Standard Model was in use in a wide variety of language implementations.

In the mid-1960s software overruns were of such concern that efforts to restrain error propagation, which essentially meant controlling the formation and use of addresses, were under way. It was clear that a solution to the problem could be found in the Standard Model provided the structural components were strictly observed. And if intra-program protection could be achieved, as it was in the tree formulation, inter-program protection followed.

R1 was susceptible to failure in store management because of inability to discriminate safely between addresses and data in the central processor, and although ways were found to “work round” the problem it was necessary to find a safe solution in any commercial time-sharing environment. Barton’s first machine, the Burroughs B5000, had similar problems of security. Adding tags to all memory words in the Burroughs B6700 and copying them securely through all data paths solved the problem. But experience of R1 had shown that only a small fraction of memory needed to be marked by tags, and it was unnecessary to go to the expense of tagging every word.

The redesigns, which started in 1963, resulted in the Burroughs B6700 family and Ferranti’s (later ICL’s) experimental Basic Language Machine (BLM). The two new architectures were demonstrated in 1968 (or thereabouts – it is difficult to be more precise). The B6700 has been described many times in open literature. The BLM is briefly summarized in [Basic Machine Principles, Macdonald-Elsevier 1972], but detailed specification appears only in ICL internal documentation.

At the insistence of D. W. Davies, who was overseeing the SERC sponsorship of advanced computer techniques, two teams were assembled in ICL: one to put the machine into operation, the other to evaluate it independently. That was probably a unique approach in the context of product development but not unreasonable since what was proposed was indeed radical and not a predictable tweak of an already existing machine. R.W. Mitchell was in charge of hardware build, J.J.L Williams and the software team designed language and system, and R. V. Jones directed the evaluation. G. F. Coulouris conducted further evaluation at Imperial College.

Architecture

Program Storage

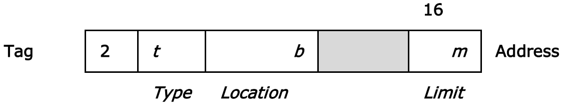
The BLM program store is conceived as a collection of ordered, finite sets of information, the length of each set being chosen as required by the programmer. Associated with each is a unique codeword. Besides giving the length of the set, the codeword indicates its physical starting position and the sort of element it contains. Usually the elements of a set are of the same sort, characterised by a numerical coding of type (integer, instruction, codeword, etc.), size (8, 16, 32 or 64 bits) and protection (read-only) status. The opportunity was taken to minimise memory fragmentation by compacting fixed structures using “relative” codewords.

It is readily seen that the program store can assume the form of a “tree”, provided certain conventions are adopted to prevent circularity. The store is thus presented as a hierarchy, with codeword sets at the intermediate levels, and instructions or data occurring at the “lowest” level. The “top” consists of a set of codewords in fixed locations, describing both permanent and variable system information.

One level down is a set of codewords describing the process bases, one base being associated with each active instruction stream. The user is given free access only to one base and to the lower level information that it describes. The machine and system functions do not allow progress "up" the hierarchy and so from one base to another: in that way inter-program protection is achieved. A codeword not defining a data sequence can be put to other uses: it can be identified as a file, device, process, secondary storage reference, and so on.

In operating on the tree structure two related problems have to be solved. Firstly, repeated travel through codewords to gain access to data multiplies memory activity. Secondly, having gained access to data the temptation to retain an address to select an "adjacent" element is practically irresistible. The consequence is seen in resort to assembly code or, in more modern terms, in the introduction of pointer variables: in either case the structure is open to violation.

The solution was to introduce a limited address, which is formed when a codeword is evaluated into a register. Such an address contains the location, limit, and type, and it is marked as such by a tag: this is the only situation in which tags are essential. Functions are provided to modify the starting location or to reduce the limit, thus forming a new address referring to a proper subsequence of the original, and to fetch or store the value of the first element. A user cannot treat an address as numerical data, so it is unnecessary to disclose its format.



Address Fields in BLM

To complete the storage picture, mixed elements each of 64 bits plus tag are introduced. In memory (i.e. in mixed sets) a 32-bit word was appended to hold the tag. The ability to represent addresses in the above way was of immense importance. It enabled the architecture to support direct access to data without endangering security. In this way intra-program protection was used to establish any required partition of program space — program environment, process, procedure, parameter list, and so on — with far greater accuracy than could be attained on contemporary machines (such as the ICL 2900) using segmentation and multiple "protection levels".

Registers

Register storage consisted of eight mixed elements. The tag value is determined when a register is loaded from memory. It is interrogated in the course of function execution. In the simplest representation there are just four categories of register:

(1) Numerical data on which a conventional set of arithmetic functions is defined.

(2) Null registers which are strictly meaningless as operands, and cause an error to be signalled by most machine functions. They can be set and tested by program, and may be used to lead into special system or user-defined actions.

(3) Addresses, referring to a sequence of elements in store.

(4) Control pointers, referring to an instruction stream.

In the experimental machine the numerical data was further classified as integer or floating point.

The current register contents define precisely the environment accessible "by inheritance", i.e. in the context established by previous instructions. In the initial BLM system one Mixed set was associated with each active process as a procedure activation stack, managed in such a way that private data could be saved and not leaked across procedure calls.

The only other means of gaining access to data was "by construction", that is to say by right established and verified in the course of generating code. This would allow, for example, access to common system functions at the top level of the hierarchy. It also allowed for the interconnection of separate program modules through the shared program base, i.e. a set of codewords giving access to instructions and data.

A consequence of defining the architecture symbolically was that certain registers could be reserved for the uses noted above. Thus the current control pointer, the stack pointer, and the Base address were part of the register set but not available for general use.

Instructions

Instruction formats, like addresses, are undisclosed. The main purpose of the "Basic Language" is to provide an abstract function definition, and to filter out illegal operations.

A simple R-R instruction is interpreted according to the type or types of data presented as operands. Failure to present valid data leads to immediate termination and hence early detection of programming error.

A single arithmetic function has multiple interpretations, taking into account integer and floating-point tags. Moreover, in the BLM if either operand addressed a data value of valid type it would automatically be fetched to the arithmetic unit and, if appropriate, stored afterwards. That added to the complexity of decoding and, in retrospect, was a move in the wrong direction. On the other hand, the inclusion of size and protection as part of the type coding proved a valuable facility.

The addressing instructions (such as MOD and LIM) failed when the modifier exceeded the limit found in address, but rather than abort at that stage it was more informative to return a Null result. At the same time the Invalid control condition was set: that could be tested (VA or IA) as a criterion for taking corrective action.

Assessment

The evaluation task was to implement subsets of standard commercial languages (COBOL, FORTRAN, and the interactive language JOSS) and to report on the experience gained. The overall objectives were to assess the proposals made in terms of program efficiency, operating characteristics, coding and debugging costs, and system overheads, and to consider the effect on the above of variations in component performance.

In any machine with controlled address formation one or more "flat" memories can be provided alongside codeword structures. The two architectural models can run in parallel - or even overlap - without difficulty: exactly that was done when implementing FORTRAN and COBOL. Even in that limited context many advantages accrue from the use of tagged memory: compilers, system and library procedures are protected, errors are quickly detected and reported, foreign imports (such as channel controllers) can be strictly confined to "safe" areas, and so on.

Regrettably, and unforgivably, the test vehicle was built from unreliable components, with the result that much time was spent using a simulator. Nevertheless, the task was completed, the machine ran for brief periods, and the assessment team was able to report (in September 1969):

> *"The conclusion of machine users is that the characteristic features of the BLM provide advantages that considerably outweigh any disadvantages. It has been shown that these features can be provided economically. At the same time, a desirable amount of flexibility in implementation has been retained."*

The experiment was terminated by ICL in December 1969.

Refinement

Any processor architecture targeting high security goals must satisfy certain practical aims to be successful: it must be foolproof, cheap, fast, flexible and easy to use. The above points were self-evident from the outset.

At the time when the BLM was being wheeled into the mortuary interest in Capability machines sparked into life. A number of experimental and commercial designs explored this area, though it would be hard to find any that was cheap, fast and flexible. By 1982 it was reported (Wikipedia): "iAPX432 capability architecture had now started to be regarded more as an implementation overhead than as the simplifying support it was intended to be". Experience seemed to show that however much microprogram or (in later years) semiconductor area was devoted to the problem the practical goals would still be out of reach. To make sense of this inaccurate and negative engineering picture it was necessary to revert to first principles.

The Basic Language, which had served its purpose in defining the architecture, became increasingly out-dated in comparison with current languages, such as C. Accordingly, a revised syntax for a language P was adopted with conventional forms of expression but augmented by two infixed addressing operators (' and ^ corresponding to MOD and LIM in Basic Language) for selecting sub-sequences. Fetch and store were indicated by postfix "dot" and "=." respectively. By the end of the 1970s a revised and simplified definition of the Basic Machine architecture was adopted, emulated, and used for feasibility studies. That was called the "Pointer-Number machine", or "PN Machine".

To give a taste of the style of programming followed in P, consider summing the first 100 elements of the **DATA** block in the above example of the Standard Model.

Two registers, **x** and **y** are required. Assume the codeword **C** is in the program base. Then:

```
x = C.'p.^99;   /* forms in x the address of the first
   100 elements, if they exist, else report Error */


y = 0; do {y+x.; x'1} while VA;  /* Forms the sum in y,
   leaves x NULL */
```

By paring away inessential functions it was possible to produce a reduced instruction set in which the validity of an instruction could be determined in a single logical stage by reference to the tag or tags of its operands. In many cases the tags are irrelevant. Comparison is by no means straightforward, as the above example shows. In any case, to achieve protection by any other means would entail far more linguistic intervention.

Having established at least parity with RISC processors it was of interest to examine features of controlled address formation that might be exploited to improve performance or reduce cost. There are at two, each taking advantage of the fact that addresses are explicit. Because formats are concealed they have no impact on addresses as seen by users.

Paged memory.

2	t	p	b	q	limit

Paged Address

In normal practice a virtual address is partitioned into two components, a page index p and a word or byte offset b within the page. When memory is accessed p is converted into a real page index q by reference to a stored Page Table. To avoid further references an associative cache memory is maintained, containing recently used Page Table entries, and that is searched to find p and hence q on each memory access.

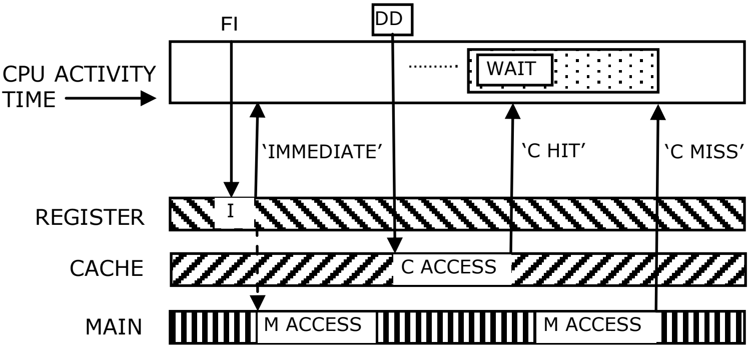
In tagged addressing it is possible to find q by reference to the Page Table as before when a register is loaded, but to retain it in the tagged part of the address. Further reference is only necessary when p changes as a result of modification. Since the Page Table is now consulted relatively rarely it can be argued that the page-associative cache and its control logic could be dispensed with. The consequences of doing so were investigated in the course of BLM assessment, which indicated that, without depending on an associative cache, memory references increased by about 15%. There are some challenging addressing patterns, but that is true in any paged system.

Forward-Looking cache memory

2	t	A	b	I	limit

Cached Address

The second opportunity arises in consideration of data caches, leading to the concept of the Forward-Looking cache.



When an address is formed in a register (at FL above) it is highly likely, though not certain, that a read or write instruction is about to be issued, affecting the area of memory to which it refers. Hence the immediate fetch of that area to fast access store would enable subsequent references to be served with minimal delay

In simplified terms it works as follows. We suppose the data cache consists of N blocks of words copied from memory, where N is at least as large as the number of registers, but not excessively so. The address A of each block is retained in the cache, together with status information indicating current activity. When a new address is formed (as above), the data cache is searched associatively: either the block is already there, or space is found to read a block from memory. The index I of the cache block is retained as part of the address. Thereafter, any read or write instruction is either served immediately from the cache or it is held up awaiting completion of the read order. Without tagged addresses the cache access cannot begin until a read or write demand is made (at DD).

There are three main influences arising from the FL addressing mechanism, which can be exploited to improve response. Firstly, because of the "advanced notice" given when an address is formed, it is expected to find the datum on the way to if not actually in the cache when it is needed. Secondly, addresses in registers define a "working set" which can be used by the purging algorithm as a guide to future demands in place of LRU measurements.

And, thirdly, when control is fragmented into a set of unitary operations, such as object or packet transformations, the cold start behaviour tends to dictate response. It is then better to have a small cache that adapts rapidly to a new locality than a large one that relies on past history for its effect. That is equally true when control passes frequently from one spatial locality to another, e.g. from user to system function.

It was possible to examine the response to the FL-Cache on the PN machine emulator. The results were qualitatively as expected but not of great numerical significance. For example, wait states (compared with Demand-Driven access) might reduce by 25%, but the actual number of wait states was in both cases quite small. The potential reductions in cache size and control logic were more significant.

It would be necessary to study response in the context of modern technology on a much more extensive scale to reach a conclusion, but it is pleasant to report that, far from increasing complexity or reducing performance, tagged addressing points to areas in which conventional design could be bettered.

The PN Machine emulation ran for several years on Motorola-based workstations. Later, with the cooperation of the DAP engineers, an alternative was provided by implementing the PN machine in the DAP controller, combining SISD with SIMD processing. That led to a third-generation descendant of R1 – the Array and Sequence Processor. Eventually, owing to changes in host architectures, maintenance became impractical. Work was discontinued in 2005. By that time it had achieved all that was intended.

Back to the future

Computer architecture as now understood can be traced back to the original formulation of EDVAC at the Moore School, and von Neumann’s exposition thereof: although many advances in engineering have been witnessed the basic structure of individual programs has changed very little. It was the intention of the Basic Machine to provide an engineering definition at a similarly fundamental level, and to study its development.

On the BLM it was shown that many advantages accrue from the use of tagged memory: compilers, system and library procedures are protected, errors are quickly detected and reported, foreign imports can be strictly confined to “safe” areas, and so on.

In addition, there were indications that, apart from solving issues of protection, other design areas would benefit from tag-dependent operations that are not available in conventional processors. They include object management, abstraction, language design, compiler technology, and operating characteristics, which are best approached by restarting from first principles.

It might have been hoped that systems would develop to take advantage of the opportunities opened up by this fundamental shift. In fact that has happened only rarely, and never more freely than on the original R1. As Perlis remarked: “Adapting old programs for new machines usually means adapting new machines to behave like old ones”. So the opportunity was lost.

Postscript

The manager of Research and Advanced Development in ICL, Gordon Scarrott, gave unstinting support to the BLM experiment, for reasons he outlined in *Resurrection 12 (Summer 1995)*. The grim commercial outlook at the time is well documented in *Resurrection 35*.

John Iliffe led the Basic Language Machine project from beginning to end. Now retired, he can be contacted at j.k.iliffe@btinternet.com*.*

A Search through the Bins of History

Iann Barron

Since I received a passing mention in the recent lecture *Proposed Computers: Several Design Projects that never left the Drawing Board* by David Eglin[6], I would like to make a couple of comments.

Choosing which comes first of two asynchronous signals is a problem for digital computers and means that all clocked digital systems are fundamentally error prone, since at some point they must communicate with an asynchronous external world.

David Wheeler (my hero at Cambridge) and I held a patent for a synchronising flip flop (circa 1970). This provided a completion signal to indicate when the flipflop had reached a stable state. The problem is that if the sampling signal and the sampled signal are nearly coincident, there is not enough gain in the circuit and the flip flop enters a metastable state. However, providing there is no significant noise, the state is genuinely metastable, and does not oscillate. It is possible, therefore, to generate a completion signal when the output has reached, say, 75% of its nominal value.

The synchronising problem existed notionally on the Modular One, although we never found any evidence that it caused a problem. However, it was a real problem for the Modular Two — another excellent computer that never saw the light of commercial day. Each module (processor, store, disc controller etc) was a single board, and a complete system was connected by a high speed bus with a 20nsec cycle time (actually, we never did better than 22nsec). Hence the need for a synchronising flip-flop.

When we designed the transputer, we tried to get people to use a single clock for multiple transputer systems, although the tolerance on the clock frequency was tight enough for this not to be essential. However, there were delays in the transmission of a clock to multiple transputers, and delays in the transmission of signals over a communication channel. As a result, each communication channel in the transputer had a synchronising flip-flop for safety.

[6] See *tinyurl.com/gksuwla* between 6 and 12 minutes in.

40 Years Ago ….
From the Pages of Computer Weekly
Brian Aldous – TNMoC Archivist

Word processor 32 launched: As expected, IBM is to introduce a video word processing system based on the System 32 small business computer, in November. Developed jointly by the general systems and office products divisions, the equipment will be a stand-alone unit aimed initially at the US marketplace. (CW502 p7)

East German machines could go on sale in UK: A range of small, general purpose computers from the main East German computer manufacturing organisation Robotron could soon be on the market in the UK if negotiations between the East German export agency, Buromaschinen-Export and a number of potential distributors are concluded successfully. (CW502 p11)

Processor launched by DEC: The dual processor DECsystem unveiled in the US earlier in the year by Digital Equipment, christened the 1088, has been formally launched in the UK. (CW502 p18)

Europe network ready on time: The world's first general purpose international computer network to be funded at government level is now up and running in Europe. The five nodes of the European Informatics Network have been connected and government organisations and higher education establishments in the UK, France, Switzerland and Italy are now linking their mainframes to the network. (CW503 p1)

British Library database plan: A group of internationally recognised databases of medical, scientific and technical information is to be made available for the first time on-line on a computer in the UK by the British Library. Many of the databases are currently run only on computers in the US, and when the first, covering medical literature, goes live in the UK in October, terminal users will be able to make massive savings in communications costs, while a much faster turnaround will be offered to customers who now submit file search requests as batch jobs. (CW504 p1)

ICL 2900 spin-off: An interesting software spin-off has come from the ICL 2900 hardware development programme and could prove useful both for tests on existing equipment and for the design and testing of new hardware. The new system, originally developed for checking the 2900 General Peripheral Controller, involved software simulation of a complex digital component. The hardware is simulated as an assembly of logic gates. Up to 7K such gates can be interconnected to form a component model — far more, says ICL, than could be simulated under previous techniques. (CW504 p9)

Enthusiasm grows for Viewdata: The theoretically unlimited quantity and diversity of information which could be made available in the home and office by the Post Office's experimental Viewdata service, has excited enormous interest in the project from both information gathering organisations and the television industry. (CW505 p4)

Coral 66 selected in the US: A striking inroad for Coral 66 into the US market has been made by Honeywell. With the approval of prominent US users, including the Department of Defense, the company is using Coral in preference to assembly language in projects for these users on its System 700 minicomputer. (CW507 p1)

Clocking-in goes automatic: With eight years to go to 1984, what many consider as the ultimate Big Brother threat is already with us. The clocking-in and clocking-out process has been "computerised" by International Time Recording. (CW507 p4)

Open University to offer time sharing facilities: A surprising newcomer to the time sharing bureau market is the Open University. It is offering time on three of its Hewlett-Packard 2000 computers at London, Newcastle and Milton Keynes for £3 an hour. (CW507 p7)

Data entry by light pen: One consequence of the widespread deployment of computers is that more and more people are required to take an active part in the operation of a data processing system. It is no longer adequate to leave the problem of feeding the computer with correct data to a data preparation department which, even now, often accounts for 30 per cent of the total cost of data processing. (CW507 p23)

GEC development encourages Intel: A software development from GEC will boost the Intel microprocessor market, especially in the UK. The company's Semiconductor division has released a resident Coral 66 compiler for the Intel 8080. This is probably the first commercial compiler for a standard high-level language to run on a microprocessor. (CW510 p8)

HP's first floppy discs: A major enhancement to the capabilities of the 9825 programmable desktop calculator has been made by Hewlett-Packard with the introduction of two floppy disc units, the first to be sold by the company with any of its products. (CW510 p10)

Forthcoming Events

London Seminar Programme

22nd Sep 2016	Crises and Lessons from the History of Software	Martyn Thomas
17th Oct 2016	Marian Rejewski's Cyklometer	Jerry McCarthy
17th Nov 2016	Sale Award Ceremony + Presentation from the winner	
15th Dec 2016	Computer Films	

Note that London meetings take place at the BCS in Southampton Street. Covent Garden starting at 14:30. Southampton Street is immediately south of (downhill from) Covent Garden market. The door can be found under an ornate Victorian clock.

You are strongly advised to use the BCS event booking service to reserve a place at CCS London seminars. Web links can be found at *www.computerconservationsociety.org/lecture.htm*.

For queries about London meetings please contact The CCS secretary Roger Johnson (page 39 for contact details).

Manchester Seminar Programme

20th Sep 2016	Simulation Models of Historical Computers — Atlas, MU5 and CDC 6600	Roland Ibbett
18th Oct 2016	Birth and Death of the Basic Language Machine	John Iliffe
15th Nov 2016	The Unknown Alan Turing	Dermot Turing

North West Group meetings take place in the Conference Centre at MSI — 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 at in the events calendar at *www.bcs.org.*

Museums

MSI : Demonstrations of the replica Small-Scale Experimental Machine at the Museum of Science and Industry in Manchester are run every Tuesday, Wednesday and Sunday between 12:00 and 14:00. Admission is free. See *www.msimanchester.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 : Colossus Galleries open daily 10.30-17.00; full Museum open Thursday, Saturday and Sunday 12.00-17.00. Situated on the Bletchley Park campus, TNMoC covers the development of computing from the "rebuilt" Colossus codebreaking machine via the Harwell Dekatron (the world's oldest working computer) to the present day. From ICL mainframes to hand-held computers. Please note TNMoC is independent of Bletchley Park Trust and there is a separate admission charge. Visitors do not need to visit Bletchley Park Trust to visit TNMoC See *www.tnmoc.org* for more details.

Science Museum :

There is an excellent display of computing and mathematics machines on the second floor. The *Information Age* gallery explores "Six Networks which Changed the World" and includes a CDC 6600 computer and its Russian equivalent, the BESM-6 as well as Pilot ACE, arguably the world's third oldest surviving computer. Other galleries include displays of ICT card-sorters and Cray supercomputers. Admission is free. See *www.sciencemuseum.org.uk* for more details.

Other Museums :

At *www.computerconservationsociety.org/museums.htm* can be found brief descriptions of various UK computing museums which may be of interest to members.

Committee of the Society

Chair: Rachel Burnett FBCS:	*rb@burnett.uk.net*
Secretary: Dr Roger Johnson FBCS:	*r.johnson@bcs.org.uk*
Treasurer: Dr David Hartley FBCS CEng:	*david.hartley@clare.cam.ac.uk*
Chair, North West Group: Prof. Tom Hinchliffe FBCS FIEE CEng:	*tah25@btinternet.com*
Secretary, North West Group: Gordon Adshead MBCS:	*gordon@adshead.com*
Resurrection Editor: Dik Leatherdale MBCS:	*dik@leatherdale.net*
Website Editor: Dik Leatherdale MBCS:	*dik@leatherdale.net*
Meetings Secretary: Kevin Murrell MBCS:	*kevin.murrell@tnmoc.org*
Membership Secretary: Dave Goodwin MBCS:	*dave.goodwin@gmail.com*
Media Officer: Dan Hayton MBCS FRSA:	*daniel@newcomen.demon.co.uk*
Digital Archivist: Prof. Simon Lavington FBCS FIEE CEng:	*lavis@essex.ac.uk*

Museum Representatives

Science Museum: Haileigh Robertson:	*haileigh.robertson@sciencemuseum.ac.uk*
National Museum of Computing: Victoria Alexander:	*victoria.alexander@tnmoc.org*

Project Leaders

SSEM: Chris Burton CEng FIEE FBCS:	*cpb@envex.demon.co.uk*
Bombe: John Harper Hon FBCS CEng MIEE:	*bombeebm@gmail.com*
Elliott: Terry Froggatt CEng MBCS:	*ccs2@tjf.org.uk*
Software Conservation: Dr Dave Holdsworth Hon FBCS:	*ecldh@leeds.ac.uk*
Elliott 401 & ICT 1301: Rod Brown:	*sayhi-torod@shedlandz.co.uk*
Harwell Dekatron Computer: Delwyn Holroyd:	*delwyn@dsl.pipex.com*
Computer Heritage: Prof. Simon Lavington FBCS FIEE CEng:	*lavis@essex.ac.uk*
ICL 2966/1900: Delwyn Holroyd:	*delwyn@dsl.pipex.com*
Analytical Engine: Dr Doron Swade MBE FBCS:	*doron.swade@blueyonder.co.uk*
EDSAC: Dr Andrew Herbert OBE FREng FBCS:	*andrew@herbertfamily.org.uk*
Bloodhound Missile/Argus: Peter Harry:	*pdh@imtex.co.uk*
IBM Group: Peter Short MBCS:	*peters@slx-online.biz*

Co-opted Members

Peta Walmisley:	*peta@pwcepis.demon.co.uk*
Prof. Martin Campbell-Kelly FBCS CITP FLSW:	*m.campbell-kelly@warwick.ac.uk*

Resurrection is the journal of the Computer Conservation Society.
Editor — Dik Leatherdale
Printed by — BCS the Chartered Institute for IT
© Computer Conservation Society