

```

procedure euler (fact, sum, eps, tim); value eps, tim; integer tim;
real procedure fact; real sum, eps;
begin integer i, k, n, t; array m[0:15]; real mn, mp, ds;
i := n; t := 0; m[0] := fact(0); sum := m[0]/2;
nextterm i := i + 1; mn := fact(i);
  for k := 0 step 1 until n do
    begin mp := (mn + m[k])/2;
      if (abs(mn) < abs(m[n])) then
        begin ds := m[n]  $\wedge$  (n < 15) then
          sum := sum + ds;
          if (abs(ds) < eps) then t := t + 1 else t := 0;
          if t < tim then go to nextterm
        end
      end
    end
  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 61**

### **Spring 2013**

### **Contents**

<b>Society Activity</b>	<b>2</b>
<b>News Round-Up</b>	<b>6</b>
<b>They Seek Him Here...</b> <i>Dik Leatherdale</i>	<b>7</b>
<b>Atlas in Context</b> <i>Simon Lavington</i>	<b>10</b>
<b>Elliott 903 Software</b> <i>Andrew Herbert</i>	<b>19</b>
<b>Dekatron – The Story of an Invention</b> <i>John &amp; Robin Acton</i>	<b>29</b>
<b>Book Review: Early Home Computers</b> <i>Dik Leatherdale</i>	<b>34</b>
<b>Forthcoming Events</b>	<b>35</b>

---

## Society Activity

---

### ICT 1301 — *Rod Brown*

On the last day of working on 6<sup>th</sup> December, the system did well and ran for a good four hours while it was demonstrated to a selection of hardware engineers, software engineers and programmers from its past. At 15:00 the machine was turned off for the last time. A meeting of these individuals in a hostelry afterwards was cheered to hear that the machine will be dismantled and transferred into a number of containers for shipment to its new home. In an ideal world this will be with the owner, however this could be to another destination if things change.

The reunion and the 50 year celebrations are now on hold as the owner has had to sell the farm which houses the machine and the project.

The project has reviewed its targets and achievements in the light of this news and although we have not been totally successful in completing all of its targets we are happy to note that the system has a loyal following of people who were directly or indirectly involved with this historic piece of British hardware. We still estimate a 95% completion of the system restoration, including the one hundred and twenty print position lineprinter.

Whatever happens now, the website for the project will remain and all of the published detail will continue to be added to. So pop back and check at [ict1301.co.uk](http://ict1301.co.uk) and the CCS site to see the latest news.

### Software Preservation — *David Holdsworth*

I have started to collect contacts for a possible team to resurrect software for LeoIII. A skeleton implementation now exists. The plan of action is at [sw.ccs.bcs.org/leo/leo.html](http://sw.ccs.bcs.org/leo/leo.html). Could it be that the LeoIII Intercode Translator is the first language processor to be written in its own language?

I recently had an idea for offering execution of our preserved software on a webserver. It is now operational, but may need to be moved to a more potent machine than my Raspberry Pi.

I would welcome feedback from members. Good to report that the EDSAC emulator dated 10<sup>th</sup> Sept 1995 on the Manchester FTP archive compiled and ran with no modification beyond changing all the filenames to lowercase. Martin Campbell-Kelly's simulator for the Wintel PC is much more elegant. However, the success of the simple 1995 version shows the value of emulation software as a way of long-term preservation of knowledge of machines of yesteryear.

## **EDSAC Replica — Andrew Herbert**

The project is on the threshold of building the first parts of the replica marking a transition from the research phase through to development.

Teversham Engineering is now making chassis types 01, 02, 03 and has completed the CAD for the racks so that as soon as chassis are wired up and tested we can start to erect the machine. We are working with Jill Clarke at TNMoC to create a suitable space for the finished replica in the Large Systems Gallery. In the interim much of the work is being done in garden sheds, garages and on kitchen tables!

Chris Burton has updated a number of our "Hardware Notes" giving us a more complete map of the machine and a library of circuit elements and construction principles we can draw on as we explore further areas of the machine. Mike Barker, chairman of the British Vintage Wireless Society, is winding coils to our specification and we have now built and tested a "lumped delay line" satisfactorily. We also have a large supply of tag strips, although the spacing of the holes is not as accurate desired. We will have to drill out the mislocated fixing points and fit washers instead. Peter Lawrence has made some improvements to his design for the Clock Pulse Generator. John Sanderson and I have produced a schematic for the Digit Pulse Generator. New volunteer Nigel Bennée has just completed a design for the half adder. Bill Purvis is looking at how to modify a Creed teleprinter to operate in the EDSAC code. Andrew Brown has started to investigate uniselectors and loading the initial orders. John Pratt has most of the store addressing system worked out. Peter Lington has been busy experimenting with the transducer for his design for nickel delay lines. We now have a display exhibit at TNMoC comprising a glass tower and case with original artefacts from EDSAC and some examples of the components and chassis we'll be using to build the replica. The tower has poster panels giving an outline of the project, EDSAC and Wilkes and his team.

Our website [www.edsac.org](http://www.edsac.org) has been much improved by Alan Clarke, working with Stephen Fleming, TNMoC PR manager. We have a growing number of videos, thanks to Dave Allen. He is producing YouTube shorts for the general audience and longer documentaries recording the project history and participants. Our overview video has had over 17,000 views. Recent radio interviews about the project have bought four new volunteers to the project all keen to over help with the construction phase.

A volunteers meeting at TNMoC is scheduled for 18<sup>th</sup> March 2013 to kick off the construction phase in earnest.

## **ICL 2966** — *Delwyn Holroyd*

Since the last report we had a fault in the DCU which was preventing boot. This was traced to the P270 board, which was then swapped out from spares. In fact both the original and replacement boards have already been repaired. The machine now boots again but fails one of the CUTS tests – more investigation needed here. The other unexpected result was a six hour run of the Primitive Level Instruction (PLI) tests without crashing, a record by some margin! Whether this is down to chance, environmental factors or having disturbed something related to the issue whilst fault-finding the DCU, is unknown at the moment.

## **Harwell Dekatron** — *Delwyn Holroyd*

Restoration is now complete and we have moved into the maintenance phase. We are going to fit a mod to the machine to allow a whole store unit of Dekatrons to be “spun” at once – this will be needed periodically to prevent the Dekatrons from becoming sticky



due to spending a long time with the glow resting on one cathode. The machine is now being run several times a week for education group visitors as well as demonstration to the public, and so far reliability has been good. We are also training up more operators.

We've found time to write a few programs, the most ambitious being one to find prime numbers. It takes 5½ hours to calculate and print the first 20 primes – not very impressive for speed but a very good test of the machine!

The Reboot event was held in November and was a great success. The machine behaved flawlessly on the day, even being put through its paces for three groups of visiting schoolchildren prior to the event! We were pleased that so many significant people from the machine's long life were able to attend, including the two surviving designers Dick Barnes and Ted Cooke-Yarborough, and it was an honour to be able to show them the machine fully working once more.

## **Bombe — John Harper**

Plans to move the Bombe Rebuild into a much larger area involving Huts 11 and 11a are starting to come together. These huts are where the small number of Bombes that were retained at Bletchley Park were housed. Hut 11 was the first heavily built hut but later versions of the Bombes (as ours is) were operated in the much larger Hut 11a.

GCHQ historic branch has invited our team to a third challenge. They will be located at ExCeL London and be part of the *The Big Bang: UK Young Scientists and Engineers Fair*. This is to take place on the 14<sup>th</sup> – 17<sup>th</sup> March 2013 inclusive.

Our operator and demonstrator training has improved to the point where we are close to being able to mount demonstrations on almost every day of the week. With so many people now involved we have found it necessary to produce a more formal system of defining operator competence with four different levels of achievement. This now allows the Bletchley Park staff who co-ordinate the volunteer activities to put together teams for a given day that consist of fully trained volunteers together with those receiving on the job training.

## **Elliott 803/903 — Terry Froggatt**

The elusive fault on one of the MX66 boards in the Elliott 903 at TNMoC, which was reported in *Resurrection 59*, has been fixed. This board failed at TNMoC in April last year, but when I took the board home, it immediately worked in my own ARCH 9000. Each MX66 board contains three balanced amplifiers that sense the core store outputs, and one was not perfectly balanced. "That'll be the problem" I thought, so I carefully adjusted the three balance potentiometers.

Back at TNMoC in June, I tried the MX66 in their 903 again. It failed again. Back at home again, it worked again. But after some weeks it did fail at home. I soon tracked down which of the three lanes was faulty, and found one transistor lead, which was broken adjacent to its can, but was still making some contact (in Hampshire but not in Bletchley). The transistor was replaced and the board is now working in the 903 at TNMoC.

We have also made some safety improvements to the 903. When it was reassembled at TNMoC a couple of years ago, we found a small ventilation grille which fitted at the left end of the paper tape controls, but we didn't fit it because we could not find the matching right end grille. It's since dawned on us that the purpose of this grille might be to keep small fingers away from a live mains terminal block in the left hand end of the desk top. So we've fitted it. There is no need for one at the right hand end.

---

## News Round-Up

---

In *Resurrection 55* we reported that the world's last manufacturer of typewriters had ceased production in India. It now seems that news of the death of the typewriter has been greatly exaggerated. In November the last typewriter to be made in Britain was presented by the Brother Company to the London Science Museum (see [www.bbc.co.uk/news/uk-20391538](http://www.bbc.co.uk/news/uk-20391538)). Even this, however, is not the end as the company continues production in the Far East. Let's wait and see.

101010101

The British Society for the History of Science Outreach and Education Committee has announced that first prize in the 2012 Great Exhibitions competition for large displays has been awarded to the Science Museum, London, for *Codebreaker: Alan Turing's Life and Legacy*.

101010101

In *Resurrection 59* I may have unwisely suggested that the seven blue plaques commemorating Alan Turing might be considered a bit over the top. Readers may have noticed that English Heritage has now suspended the scheme entirely. So the prospects for an eighth here in Teddington aren't good.

101010101

Once again to the local cinema to see *Quartet* the new film which marks the directorial debut of Dustin Hoffman. The film depicts the struggle of the residents of Beecham House, a care home for retired musicians, to stage a concert to celebrate the birthday of Giuseppe Verdi (1813–1901).

The part of Beecham House is played by Hedsor House, a location which will be familiar to many old ICL hands as TAP01.

101010101

Sad to report that, not long after the relaunch of the Harwell Dekatron computer (see next page), Ted Cooke-Yarborough passed away in January. As well as his creation of the Dekatron machine, he also designed CADET, claimed as the first fully-transistorised computer in the world and was one of the driving forces behind the early manoeuvrings for a large-scale machine – a project which was eventually realised as Atlas.



---

## They Seek Him Here....

### *Dik Leatherdale*

---

One of the many perks of the job of editing *Resurrection* is the occasional invitation to attend an event relating to the history of computing. The winter months have been particularly fruitful with no less than three visits accomplished

I *should* arrive at these events in a grubby mackintosh, notebook in hand and a dog-eared card bearing the word "Press" tucked into the hatband of my trilby. In practice, it isn't quite like that but...

### Harwell Dekatron Re-launch

So firstly to the National Museum of Computing in November for the formal re-launch of the Harwell Dekatron computer (a.k.a. WITCH). Regular readers of *Resurrection* will be familiar with the story of how this venerable piece of hardware was rediscovered in the store of the Birmingham Science Museum in 2008 and has been slowly brought back to life at TNMoC over the last few years.

The formal switch-on was something of an event with Ted Cooke-Yarborough and Dick Barnes, designers of the 61 year-old machine both giving short talks about their experiences, following Kevin Murrell who told the unlikely tale of the machine's survival in the face of no less than three previous "retirements".

Bart Fossey, a long-time user of the machine at Harwell, also spoke and re-enacted a race between the machine and a hand-cranked calculator which was alleged to have taken place nearly 60 years ago. The result was declared a draw to great acclaim. But Bart denied that such a race had, in fact, taken place, it merely having been observed that the machine was around the same speed as Bart doing a manual check on the results. The advantage lay not with the speed of the processor, as with its ability to carry on for hours, days even, at a time without getting tired or making mistakes.



**Bart Fossey races the machine**



The final guest speaker was Peter Burden who met the machine as a schoolboy during its time at Wolverhampton. Peter it was who featured in this famous *Express and Star* photograph.

The machine has now been formally recognised as the oldest working computer in the world by Guinness World Records.

The privilege of running the first program was deservedly accorded to Delwyn Holroyd who, along with Johan Iverson, Eddie Washington and Tony Frazer formed the core restoration team, to whom all credit is due. The short video of the first run can be found at [tinyurl.com/witchreboot](http://tinyurl.com/witchreboot). By early January it had been viewed over 1,000,000 times. A star is born! You too, Delwyn!

All this and a new booklet covering the project and the machine's history too! See [tinyurl.com/dekatronbook](http://tinyurl.com/dekatronbook).



**“Press any key to continue”**

## **Atlas @ 50 Celebration**

Then at the beginning of December to Manchester to celebrate the 50<sup>th</sup> anniversary of the formal launch of the Manchester University/Ferranti Atlas computer, in its time claimed as the fastest, most powerful computer in the world.

An audience of around 160 souls attended the centrepiece of the event, an afternoon of lectures dealing not so much with the technology as with the events which led to the building of this memorable machine and with the history of the three Atlas 1s and three Atlas 2s which were produced. No less than two Atlas 1 emulators were demonstrated and presentations were also given on the current research of the University's School of Computer Science. In this edition of *Resurrection* we feature Simon Lavington's overview of the history of the project. We plan to include several other articles associated with the celebrations in future editions.

All this was topped and tailed by a party for Atlas veterans and a formal dinner at which Sebastian de Ferranti was guest of honour. Simon Lavington and Dai

Edwards both paid tribute to the team which created Atlas, with particular mention of two who could not be present, Tony Brooker and Frank Sumner.

Dai Edwards, Yao Chen, David Howarth and Mike Wyld were interviewed in depth the following day. A professional standard audio recording has been made for the benefit of future historians.

As a minor participant in this event it is not for me to assess it, but I cannot let the opportunity pass without acknowledging everybody concerned in the event, particularly Simon Lavington who has been quietly toiling away for over a year to lead the organisation of this splendid event. Thanks too to Manchester University for their unstinting help (especially Toby Howard) and to various organisations whose support made the event possible – the BCS, the Institution of Engineering and Technology, Google and the Science and Technology Facilities Council.

## The IBM Museum

Finally in January to Hursley House, IBM's stately home in darkest Hampshire, with a small group of CCS committee members. On the lower ground floor may be found the most wonderful collection of IBM artefacts ranging from early time-keeping clocks to an extensive collection of PCs and tablets as well as a comprehensive library of computer-related books. We were shown round by a small group of retired IBM employees



**An IBM Displaywriter**

who run the museum and who expressed a wish to become more closely connected with the Computer Conservation Society – an idea we are keen to progress and on which we have started work by interlinking our two websites.

We look forward to working closely with the IBM Museum.

The Museum's website is at [hursley.slx-online.biz/the\\_museum.asp](http://hursley.slx-online.biz/the_museum.asp).

---

## **Atlas in Context**

*Simon Lavington*

---

The Ferranti Atlas deserves to be remembered for two reasons: its immediate impact on UK computing resources and its longer-term influence on computer design. The first was short-lived; the second is with us today. More on both of these later. First, it's helpful to recall the local and national motivations that led to the birth of the Atlas project. We'll then describe the sites at which the production Atlas computers were installed between 1962 and 1967 and their impact. Finally, in the light of hindsight, we'll try to assess the wider influences of Atlas.

### **The Nuclear Imperative**

The Cold War era of the mid-1950s placed much emphasis on the use of nuclear energy – both for defence and for power generation. This nuclear imperative required massive scientific computing resources. It was the perceived need for massive computing resources that became the context for the birth of Atlas.

In the USA, the nuclear imperative was recognised at places such as the Atomic Energy Commission's Los Alamos Laboratory which, in 1956, contracted IBM to design and build a supercomputer called STRETCH (aka the IBM 7030). IBM's target was to produce a performance two orders of magnitude faster than the company's current product, the IBM 704. In round figures the aim (not quite achieved) was a performance of one million floating point additions per second. The first of seven STRETCH computers was delivered to Los Alamos in April 1961. It cost \$13.5 million.

In 1956 news of the STRETCH project reinforced the UK Atomic Energy Authority's own desire for a machine of similar performance. The UKAEA quantified their requirements as a need to perform  $10^5$  three-dimensional mesh-operations per second and a primary store of  $10^5$  numbers. Each mesh-operation equated to about ten basic machine instructions, one of which was a floating-point multiplication. It seemed most unlikely that a local UK product would be available to meet this target. The early lead in computers that Britain had achieved in the period 1948 – 51 had slipped away. By the end of 1955 there were only about a dozen production digital computers in use in the country. Half of these were manufactured by Ferranti Ltd. The remainder came from the Elliott, English Electric, BTM and Lyons companies. All these British-designed computers were single-user systems with practically no systems software. They each had primary memories of 4K bytes or less and could obey only about 1,000 instructions per second. Their technology was based on vacuum tubes

(thermionic valves). In 1956 no British company had plans for a machine with anything remotely approaching a speed of one million floating-point additions per second.

From 1957 onwards, motivated by the UKAEA's needs, the National Research Development Corporation (NRDC) began to involve itself in promoting the idea of a new British high-performance computer to match STRETCH. NRDC had already supported the fledgling British computer industry, having given financial help for the development of the Ferranti Mark I\*, the Elliott 401 and the Ferranti Pegasus. In contrast, the promotion of a supercomputer project proved very much more difficult. Towards the end of 1958 Lord Halsbury, NRDC's Director, said: "Were we [the UK] strong enough to compete? Ought we to try? Could we afford not to? Could any such proposal be established on a commercial basis? During the last two years I have unsuccessfully wrestled with divided counsels on all these issues." The story of NRDC's efforts between 1957 and 1959 is a complex one, involving at least four successive plans. What follows is a simplified summary of these plans for a national supercomputer project.

Throughout, NRDC sought the advice of computer experts from the UKAEA, the Royal Radar Establishment, the National Physical Laboratory, and the universities of Cambridge and Manchester. In the spring of 1957 NRDC's Plan A was that a British supercomputer should be designed by a committee with membership drawn from these organisations. NRDC would invest £1 million of its own funds and NRDC's in-house computer consultant, Christopher Strachey, would take an active part in the design. The manufacture would be sub-contracted to British industry and the UKAEA would purchase the resulting machine.

The principal difficulty with Plan A was that it did not find favour with the UKAEA. Indeed, by June 1957 Sir William Penney, the UKAEA's senior adviser, stated that there was no prospect of UKAEA placing an advance order for a British supercomputer. Whilst admitting that the Atomic Weapons Research Establishment (AWRE) at Aldermaston would certainly need a fast computer, AWRE had to take the first to become available as a proven product, be it British or American. The case for Harwell to have a fast computer had not yet been approved.

Nevertheless, NRDC was undeterred. By this time, IBM was said to be spending \$28m annually on STRETCH and to be deploying 300 graduates on the project. NRDC was convinced that there should be a British answer to STRETCH.

Meanwhile, an independent new computer project was getting under way at Manchester University. During 1956 Tom Kilburn's research group in the

Department of Electrical Engineering had started to develop a novel parallel adder circuit using transistors and was also investigating high-speed ferrite RAM and ROM storage. By early 1957 Kilburn's group had decided to design and build a fast computer with a target speed of one microsecond per instruction. The project was called MUSE, short for Musecond Engine. To support the project, a sum of £50K was made available from the Department's own research fund. This fund had been accumulating since 1951, as an agreed percentage of the University's overall Computer Earnings Fund.

### **NRDC's Plans B, C and D**

NRDC's Plan B, devised early in 1958, was to set up a Working Party to consider MUSE as the design-basis for a British supercomputer. The membership consisted of Strachey from NRDC, David Wheeler from Cambridge University, Jack Howlett and Ted Cooke-Yarborough from the Atomic Energy Research Establishment at Harwell, and Tom Kilburn from Manchester University. The Working Party visited Manchester on 17<sup>th</sup> March. They were shown the MUSE developments, which by then included about 500 words of the MUSE fixed store (ROM) with an access-time of 0.2 microseconds, one word of a two-core-per bit RAM with a cycle time of 2.5 microseconds, and two single-bit experimental adders using different types of transistor. (There was an extreme shortage of suitable transistors at that time and none of these was manufactured in the UK. The American transistors in MUSE cost about £30 each – equivalent to £600 in today's money).

In their Report, the NRDC Working Party rejected MUSE as the sole basis for a British supercomputer, commenting that the speed of MUSE seemed more or less acceptable but the arrangement for storage and peripherals was inadequate. Therefore a "fresh logical design" was necessary. In reply, Kilburn made two points:

- (i) MUSE was not dependent upon support from NRDC and would be completed without it;
- (ii) he would only accept responsibility for participation in an NRDC-sponsored project if he was "the final arbiter on all questions of engineering and logical design".

Not surprisingly, NRDC moved on to Plan C which, by the summer of 1958, involved direct approaches to three British manufacturers: EMI Ltd., English Electric Ltd. and Ferranti Ltd. English Electric responded negatively and EMI positively. NRDC inclined strongly towards EMI, with whom Christopher Strachey had developed a good rapport. Ferranti hesitated, concerned about the method

and time-period for repayment of an NRDC loan and about who was to be the design authority for the project.

Meanwhile Ferranti was already committed to developing a large business data-processing computer (Orion) and had also kept in close informal touch with the MUSE project at Manchester University. Then in October 1958, confirmed officially in January 1959, Ferranti decided to go ahead with a high-performance project jointly with Manchester University, the project thereupon changing its name from MUSE to Atlas. It was agreed that the first production Atlas would be installed at Manchester University, which would have half the operating time, and that Ferranti would pay Manchester University £10K per annum plus a percentage of the income gained by selling time on the computer. The MUSE team of about 15 academics was increased by the addition of between 30 and 40 Ferranti employees and the production of Atlas printed-circuit boards was commenced at Ferranti's West Gorton factory in Manchester.

By March 1959 NRDC had decided "by the narrowest of margins" that Plan C should be to put all of its resources behind an EMI supercomputer project rather than supporting the Ferranti/Manchester efforts. However this decision was changed a few weeks later – more or less coincident with the resignation of Christopher Strachey from his NRDC post. For Plan D, lesser sums (of £250K to EMI and £300K to Ferranti) were loaned to each company, the loans to be recovered by a levy on the total computer sales of each firm. The EMI supercomputer project effectively came to nought in the end. Ferranti, as has been seen, was already committed to Atlas before NRDC offered the grant and was able to repay the £300K loan by 1963. Indeed, Sebastian de Ferranti had at one stage suggested that the British supercomputer project should be called BISON: Built In Spite Of NRDC.

## **Atlas arrives at last**

The first production Atlas was officially inaugurated at Manchester University by Sir John Cockcroft on 7<sup>th</sup> December 1962, though it was not until several months later that a regular computing service was running.



**Sir John Cockcroft, Sebastian de Ferranti and Tom Kilburn inaugurate the Manchester Atlas in 1962**

For more photographs of Atlas, hardware and software technical references, simulators and a growing collection of recent retrospective articles on the Atlas project, see [www.cs.manchester.ac.uk/Atlas50/](http://www.cs.manchester.ac.uk/Atlas50/).

Priced at between £2m and £3m each, equivalent to about £50m today, it was clear that Ferranti would only be able to sell a few Atlas computers. The complete list, with dates of installation and de-commissioning, is given in Table 1. The cheaper Atlas 2 model in Table 1 was a joint development between Ferranti Ltd. and the University of Cambridge. Cost was saved for Atlas 2 by removing the drums and one-level paged store and the fast fixed store (ROM) of the full Atlas 1 machine, by reducing the speed of the primary core store from 2 to 5 microseconds cycle-time, and by re-designing the tape and peripheral co-ordinator units. The Cambridge Atlas 2 was called Titan. In 1967 Cambridge University released an interactive time-sharing operating system for Titan and for the CAD Atlas 2.

Site	Installation Date	Machine Type	Switch-off date
University of Manchester	1962	Atlas 1 with 16Kwords of core plus 4 drums	1971
University of London	1963	Atlas 1 with 32Kwords of core plus 4 drums	1972
National Institute for Research in Nuclear Science, Chilton (adjacent to UKAERE Harwell)	1964	Atlas 1 with 48Kwords of core plus 4 drums	1973
University of Cambridge	1964	Atlas 2 (Titan) with 128Kwords of core	1973
AWRE, Aldermaston	1964	Atlas 2 with 128Kwords of core	1971
Ministry of Technology's Computer Aided Design Centre, Cambridge	1967	Atlas 2 with 256Kwords of core	1976

Table 1: Ferranti Atlas installations.

Ferranti made strenuous efforts to sell Atlas in America (principally to Atomic Energy establishments), in Australia (principally to CSIRO) and in Europe (principally to CERN). It was little comfort that IBM only sold seven of the rival IBM STRETCH computers. By 1964 the real threat, both to Atlas and to STRETCH, was the arrival of a new supercomputer manufacturer in America, the Control Data Corporation. The CDC 6600, which benefited from advances in

silicon transistors, was about three times faster than both the IBM STRETCH and Atlas (the figures are given later, in Table 2).

Once each of the installations of Table 1 was fully in service, the immediate impression was of their high rate of work-throughput compared with previously-available computing systems. Atlas' ability to schedule work automatically in a multiprogramming environment heralded a new generation of computing resource. The statistics speak for themselves.



**David Howarth stands by the printer on the upper floor of the Chilton Atlas**

time had been devoted to UK universities, during which 2,300 research projects had been supported. The remaining 15% of computing time was used by government departments for applications such as weather forecasting and space research. Of the 44,500 scheduled hours of computing time during the period May 1965 to March 1973, 43,000 hours had actually been usable – yielding an Atlas availability of 97%. The central processor usage statistics during this period were:

User programs:	82% of CPU time;
Supervisor (i.e. Operating System) activity:	12% of CPU time;
Idle time:	6% of CPU time.

At Chilton, as at the other sites, obtaining rapid access from remote universities to Atlas facilities was a major driving force for the development of networking in the UK – as indeed it was for ARPANET in the USA.

At the Chilton Atlas close-down ceremony on 30<sup>th</sup> March 1973 the Director, Jack Howlett, was able to report as follows: From May 1965 to March 1973, 863,000 jobs had been handled by the Chilton Atlas, with a total market value of computing time of £10.8m.

Approximately 85% of available computing

The six Atlas installations also inspired much innovative development of systems software. This took the form of new programming languages, new applications packages, and a new Operating System (the Titan time-sharing system). The Compiler Compiler at Manchester is another specific example of software innovation. Finally, some of the graphics and engineering design packages developed at the CAD Centre are still in use today.

## How powerful was Atlas?

In 1962 the art of benchmarking was in its infancy so raw speed was the measure frequently used to compare scientific computers. Most systems were still effectively single-user; concepts such as multi-tasking and workload throughput were relatively unknown. As measured by raw instruction speed, Atlas was a little slower than its rival, the IBM STRETCH. This is shown by the figures in Table 2.

Instruction	IBM 704 1955	Ferranti Mercury (1957)	IBM 7090 (1959)	IBM 7030 (1961)	Ferranti ATLAS (1962)	English Electric KDF9 (1963)	CDC 6600 (1964)
FXPT ADD	24	60	4.8	1.5	1.59	1	0.2
FLPT ADD	84	180	16.8	1.38–1.5	1.61–2.61	6–10	0.3
FLPT MPY	304	300 (360)	16.8–40.8	2.48–2.7	4.97	14–18	1

Table 2. Instruction times in microseconds for four American and three British computers that were first installed between 1954 and 1964.

By the autumn of 1962 there were 13 Ferranti Mercury computers installed in the UK, one IBM 704 (delivered in 1957, so most likely traded back to IBM by 1962 by UKAEA) and four IBM 7090s. An IBM STRETCH had just been delivered to AWRE Aldermaston. There were of course many other smaller computers in the UK in 1962 but most had no hardware floating-point and their scientific computing power was very small compared to the machines in Table 2. The appearance of the first Atlas in 1962 therefore significantly increased the potential scientific computing resources of the whole of the UK.

Actually the instruction times given in Table 2 do not tell the full story. Bob Hopgood of Chilton, who wrote compilers for both STRETCH and Atlas and

implemented a large Quantum Chemistry package (MIDIAT) on both computers, has said:

“STRETCH could run extremely fast if you had the code set up just right and it remained in core memory. It had some terrible deficiencies as well. It made guesses as to which way a conditional jump would go and if you got it wrong it had to backup all the computation it had done. So the same



**Dark satanic mill - The lower floor of the London Atlas with its rows of processor cabinets**

conditional jump could be as much as a factor of 16 different in time between guessing right and wrong. The STRETCH nuclear weapon codes at AWRE Aldermaston probably outperformed Atlas by quite a bit. On the other hand Atlas ran some large number theory and matrix manipulation calculations much faster than STRETCH. My codes were pretty similar in performance but on large calculations where intermediate results had to be stored on magnetic tape, Atlas was significantly faster due to the Ampex tape decks. I think on an untuned general purpose workload Atlas was faster and if the code was tuned to STRETCH it would be faster. In conclusion, I would say that in 1962 ‘Atlas was reckoned to be the world’s most powerful general-purpose computer’”.

This last comment echoes Hugh Devonald of Ferranti, who said in 1962: “Atlas is in fact claimed to be the world’s most powerful computing system. By such a claim it is meant that, if Atlas and any of its rivals were presented simultaneously with similar large sets of representative computing jobs, Atlas should complete its set ahead of all other computers.”

In addition to its fast arithmetic and storage facilities, the Atlas arrangements for handling peripheral devices and the Atlas multiprogramming Supervisor (i.e. Operating System) were credited with being instrumental in “setting Atlas ahead” of the others in terms of program throughput. In 1962 Hugh Devonald added: “The Supervisor is the most ambitious attempt ever made to control automatically the flow of work through a computer. Its ability to handle the varied workloads that a machine of this size tackles will influence the future

design of all computers". Per Brinch Hanson's 2001 book *Classic Operating Systems* describes the Supervisor as "the most significant breakthrough in the history of operating systems". The Atlas Supervisor consisted of some 35,000 machine instructions. Incredible though it now seems, it had been developed by a team that never numbered more than six full-time programmers.

## **The Atlas legacy**

The Supervisor certainly did exert some influence on the development of future computing systems. Perhaps more subtly, the Atlas Compiler Compiler also made its mark on Computer Science. George Coulouris, who used the Compiler Compiler in the mid-1960s, has recently remarked that he still finds the Compiler Compiler "a quite amazing achievement in terms of the innovations that it contained and the effectiveness of its design and implementation. It encompassed innovative contributions at so many levels, from the very concept of a compiler-compiler to the inclusion of a domain-specific language for applications in system programming and the combination of interpretation with code generation".

As to Atlas hardware, it is the way with digital technology that it is soon superseded. Some of the Atlas systems architecture ideas do, however, seem to have had a longer life. Two ideas that stand out are (a) Extracodes and (b) the paged One-level Store with its automatic virtual-to-real address translation. It is interesting that, in 1962, a number of computer experts (Stan Gill and Christopher Strachey amongst them) were very sceptical of the Atlas one-level store scheme. Perhaps the final comment may come from Robin Kerr, a member of the Atlas team at Manchester from 1959 to 1964 who subsequently worked for a number of American computing corporations including Control Data, GE's Corporate Research Labs and Schlumberger. Robin Kerr recently said: "The Atlas project produced the patents for Virtual Memory. I would claim that Virtual Memory is the most significant computer design development in the last 50 years. Certainly it is the most widely used".

I think Tom Kilburn, who alas died in 2002, would have been rather pleased to hear this.

*This article is an expanded version of the talk given by the author during the Atlas@50 event in Manchester in December 2012. He can be contacted at [lavis@essex.ac.uk](mailto:lavis@essex.ac.uk)*

*Illustrations courtesy of MOSI and Rutherford Lab.*

---

# Elliott 903 Software

*Andrew Herbert*

---

In September 2011, I became the owner of an Elliott 903 computer. Originally delivered by Elliotts to AERE around 1967, the machine was purchased by Don Hunter in 1978 and used by him as a home computer for many years. Now advanced in years, Don no longer wanted to the machine in his home, so aided by Terry Froggatt, I moved it to my house in Cambridge. As well as the computer Don handed over several boxes of paper tapes and manuals. After re-commissioning the machine with Terry's help, I set myself the task of archiving and cataloguing the paper tapes and documentation with the aim of establishing master copies of what a 903 user would expect to have in the early 1970s just before Elliotts ceased providing support and the machines became obsolescent.



## **The Elliott 903**

The Elliott 903 computer is described in some detail on the "Our Computer Heritage" website. In summary, it is an 18-bit minicomputer with a minimum of 8K of core store. The 903 was a civilian packaging of the military Elliott 920B computer. Designed in the early 1960s, it was a discrete transistor machine with a typical instruction time of around 25 microseconds. It was a popular teaching machine in universities and colleges where the 16K configuration was commonly deployed to allow load-and-go operation of ALGOL 60 and FORTRAN. Some industrial users further extended the memory to 24K and beyond to run programs on large data sets.

The order code is very simple with 16 basic instructions operating on fixed-point fractions: floating-point arithmetic has to be performed by software.

The primary means of input/output is eight-hole paper tape. While nominally an optional peripheral, most 903s came with an attached ASR33 teletype. Graph plotters, magnetic tape drives, lineprinters and card readers were also available (identical to those supplied with the Elliott 4100 series). Physically the basic machine resembled a modern chest freezer. Further cabinets could be added to hold additional core store, up to a maximum of 64K, and/or additional peripheral interfaces, along with additional power supplies. There was an operator's control panel that stood on top of the cabinet together with paper tape reader and punch. Many installations also had an engineer's diagnostic panel with neon displays of the principal registers etc.

While there were interrupts to allow multi-programming and asynchronous input-output, there was no virtual memory provision so most 903 computers were used as single user machines.

When new, in the late 1960s, a typical system would cost around £25,000.

### **Elliott 903 Software**

Elliott's supplied software to users bundled in with the purchase and on-going maintenance support for the computer itself: in those days software was seen by manufacturers as a cost of doing business rather than a source of revenue. The standard suite consisted of:

1. Programming language systems (ALGOL 60, FORTRAN II and SIR assembler)
2. Subroutine libraries
3. Development utilities (e.g., editors, debugging aids)
4. Applications
5. Test programs.

Notably this list does not include an operating system. Most users ran programs on the bare machine and the programming language systems were self-contained. There was a 9 KHz magnetic tape based batch operating system called FAS which I encountered as a schoolboy, sending FORTRAN programs to be run on Elliott 903 at the now defunct Medway and Maidstone College of Technology in Rochester, Kent. There was also a disc operating system called RADOS for the Elliott 905 computer, a faster and upwards compatible successor to the 903, but that is outside the scope of this article.

Elliott's distributed software on paper tape. Programming systems, utilities and applications were shipped as "sum checked binary" tapes suitable for loading using the 903's initial instructions. Libraries were supplied either as source code or an intermediate "relocatable binary code (RLB)" suitable for linking with compiled programs.

Elliott's settled on using the emergent ASCII code for source tapes, for compatibility with the 4100 series, moving away from the earlier private code used on their 503 computers. Unfortunately ASCII changed somewhat during the life of the 903 with some symbols unhelpfully swapping positions in the code table. Most 903 software essentially ignored this - a later "900 telecode" was introduced in place of the original "903 telecode" but in reality all that happened was an update to the documentation to reflect the new graphics associated with the binary codes used in the software. This was particularly tiresome for ALGOL 60 users where string quotes changed from acute and grave accents ('**a pretty string**') to quote and at sign ('**an ugly string**@), and as with many other contemporary British machines, £, # and \$ played musical chairs.

## **Programming Languages**

Elliott's provided three programming language systems for the 903: ALGOL 60, FORTRAN II and SIR (Symbolic Input Routine - i.e., an assembler).

ALGOL and FORTRAN were both provided as either a two-pass system for use on 8K machines, or as an integrated load and go system for 16K machines. In the two pass systems an intermediate tape was produced for communication between the separate passes.

In the case of ALGOL, the first pass was the compiler (called the "translator" by Elliott's) that produced an intermediate stack-based code rather than relocatable binary machine code (although the same tape format was used so that the standard 903 loader could be used to put the bits into store). The second pass run time system comprised the interpreter for the intermediate code and standard libraries.

In the case of FORTRAN, the compiler produced either SIR code or RLB code and therefore the runtime was basically just the standard loader and a set of libraries.

In both systems the user could reclaim the memory allocated to the standard library by scanning a library tape containing RLB and only those routines needed by the program would be loaded.

Both systems also provided a "large program" second pass, essentially the standard second pass modified to allow programs and data to extend beyond the 8K limit of the basic system.

The ALGOL system was developed by CAP and based on the Whetstone ALGOL system for the KDF9 developed by Randall and Russell, written up by them in their book *ALGOL 60 Implementation* published by Academic Press in 1964. Indeed having found some source tapes I discovered that the 903 ALGOL system is essentially an SIR transcription of the flow charts in that book. 903 ALGOL has a few restrictions compared to its parent: recursion and own variables are disallowed and all identifiers have to be declared before they can be used. More usefully, the language supports **PRINT** and **READ** statements as found in 803 and 503 ALGOL which are convenient to use and allow good control of output formatting.

The restriction on recursion is easily circumvented with a patch to remove the check for it in the translator. Everything is fine at run time provided the recursive procedure does not have local variables – since these are statically allocated (i.e., making them **OWN** variables....). A programmer can achieve the effect of dynamic local variables by nesting a local procedure within the recursive one. Don Hunter developed such a patch and several others to add further extensions that are embodied in his ALGOL system for the Elliott 903 simulator to be found on the CCS simulator archive. I've subsequently reverse engineered a paper tape version from this and added further patches due to Terry Froggatt that deal with the character code issue, resulting in the first new release of Elliott ALGOL 60 in nearly 40 years!

The FORTRAN II system is little more than a macro-generator: FORTRAN statements are translated line-by-line into machine code with no optimisation. While generating machine code might be thought to lead to faster programs when compared to ALGOL, the fact that both have to implement floating-point arithmetic using an interpreter erodes any difference. Moreover, when debugging, ALGOL's stricter checking of integer arithmetic for overflow can be an advantage.

Both ALGOL and FORTRAN provided facilities for writing procedures in machine code. In the case of ALGOL, and optionally for FORTRAN, these have to be independently compiled and linked. FORTRAN also allows SIR assembly code to be included inline. In both cases there are strict rules defining how machine code should be written to conform to the runtime conventions of the corresponding language system.

It was not possible to separately compile and link ALGOL procedures, using Elliott's software. In part this was because the output of the ALGOL translator was intermediate code for the runtime interpreter rather than machine code. However Don Hunter subsequently provided a means to compile ALGOL procedures independently for the Elliott 903 simulator and in principle his system could be run on a real 903.

The main assembler provided by Elliotts was called SIR (for Symbolic Input Routine). SIR was the successor to an earlier more primitive assembler called T2 that had been developed for the first machines in the 900 series (i.e., the 920A).

SIR allows the use of integer, fractional, octal and alphanumeric constants, symbolic labels, literals, relative addressing and comments as its main features. It does not provide any form of macro generation facility and has limited ability to do arithmetic on labels as addresses. Instruction function codes are expressed numerically rather than mnemonically.

T2 by contrast only accepts instructions with simple relative addressing and integer constants. While essentially obsolete at the time of the 903, T2 was given to users as a number of utilities were written in T2 or required data laid out as if a T2 program block. Many library routines were also supplied in T2 notation. By design SIR was made upwards compatible with T2 so these could be assembled in both systems.

SIR could be operated in either load-and-go mode or made to produce RLB tapes preceded by standard RLB loader. Load-and-go mode was convenient for small programs, but if the space occupied by the assembler was needed, or a self-contained binary tape for loading under initial instructions was required the RLB option was preferable. There is no support in SIR for program code to be located outside the first 8K of memory, which was to become an issue later in the history of the 900 series as 16K and larger memories became prevalent. (The work around used to assemble the 16K load-and-go systems was to build them in the bottom 8K, then run a utility program to copy them to the upper 8K. As the machine code is based on 8K relative addressing there was no need to fix up addresses, except for references between the two 8K modules).

## Library Routines

The library routines provided by Elliotts were principally input/output and mathematical functions. A double precision interpreter (QDLA) and a floating-point interpreter (QF) were provided along with mathematical functions for these formats. In the case of ALGOL and FORTRAN versions of these packages were built into the language systems: the standalone library versions were intended for SIR programmers to use.

For ALGOL programmers there was a source code matrix library called ALMAT. This package originated on the 803/503 computers and provided routines for matrix operations and linear algebra functions.

For machines with graph plotters, lineprinters and card readers there were appropriate library routines (device drivers in modern terms) for the SIR programmer. ALGOL had its own plotter library (based on the library for the Elliott 4100 series). Both ALGOL and FORTRAN integrated card reading and line printing into their higher-level input/output facilities.

The library routines were all known by short identifiers beginning with the letter Q (e.g., QSQRT, and strictly "Q not followed by U"): those which had been inherited from the early 900 series software were often also known by shorter code names such as "B1" (QLN), reminiscent of the EDSAC convention for naming library tapes.

The only data processing oriented routine was a Shellsort package for in-memory "files", and the interface to this was very basic — essentially a table defining record structure and the order in which fields were to be sorted. There was no explicit support for fields that spanned multiple words or had complex structure (e.g., floating point numbers) although a programmer who understood the machine representation of the data could work around this.

## Utility Programs

The utility programs provided by Elliotts were mostly concerned with paper tape preparation. EDIT was a simple text editor, reading in a steering tape of editing commands to be applied to a source tape and punching an updated version. It has commands for copying or deleting to an identified string, making replacements and inserting new text. Two tape copy programs were provided: COPYTAPE for short tapes and QCOPY for long tapes. The former read the entire tape into store before punching a copy, which was kinder on the tape reader than QCOPY, which operated character by character, forcing the reader to continually brake the input tape as it waited for the punch. Given the potential unreliability of paper tape as a medium all these routines provided a means to rescan the

input and read back the output to ensure misreads and/or mispunches were detected. (And as Terry Froggatt points out, the truly paranoid read back the input again after punching to confirm the absence of store corruption...)

Debugging aids comprised two utilities: MONITOR and QCHECK. Both had to be assembled as part of the system being debugged. As the name suggests, MONITOR provided facilities to monitor the execution of a program by printing out registers and memory etc. whenever the program executed one of a specified set of program locations. QCHECK provided an interactive debugging interface enabling memory to be inspected and modified and for further program execution to be triggered. QCHECK itself could be triggered by an interrupt from the operators' console. Both operated at the machine code level and were of little use to ALGOL and FORTRAN programmers.

A separate group of utilities provided a set of tools to allow the programmer to build binary tapes that would load under initial instructions (i.e., not requiring the relocatable binary loader).

QBINOUT was used to punch out short binary programs to paper tape, in a form suitable for loading by the initial instructions. Typically the punched program would itself be a more powerful loader, of which there were several. T22/23 provided just a loader (the T22 part) and the means to dump regions of store out in T22 format (the T23 part). The paper tape format was more compact than that produced by QBINOUT and included checksums to guard against errors. A further utility C4 was provided to check the contents of a punched T22/23 a tape against store. (Interestingly some internal documentation refers to T22 as QSCBIN and T23 as QSCBOUT but these names didn't make it to the 903 user documentation.) A more advanced loader was that used by the three language systems to link and load separately compiled libraries and programs help in RLB form.

## **Applications**

There were just two applications provided by Elliotts – the Elliott Simulation Package (ESP) and a PERT project planning system.

ESP consisted of a machine code library and accompanying ALGOL source code routines for generating random numbers drawn from different distributions, histogramming and setting up event based simulations. The user was expected to write their simulation using the supplied source code as a foundation, compile the result and link with the small machine code library that contained a random number generator.

The PERT application was standalone, reading a steering tape to set the system up then a series of data tapes describing individual projects to be analysed. The application would then print tables of shortest paths, earliest and latest deadline and so forth.

## **Test Programs**

Elliotts offered a large collection of test programs for the 903. Most were intended for use for fault-finding by engineers, but a few were supplied to users for use at start of day to check the condition of the machine and paper tape system. All the test tapes were named  $X_n$ , as in "X3", the processor function test.

## **Other Software**

The advent of the Elliott 905 brought a new version of SIR called MASIR which included macro generation facilities and new symbolic address formats to better enable programming across 8K memory module boundaries as 16K and larger memory configurations became the norm. Instruction mnemonics were introduced to cover the fact that the additional instructions in the 905 repertoire were all coded as function code 15. MASIR came with a new linking loader (called "900 LOADER") with a new RLB format. Both MASIR and the 900 LOADER were issued to 903 users.

A new FORTRAN IV system was provided for the 905. This provided a full implementation of the ASA standard and generated RLB suitable for the new 900 LOADER. While 905 FORTRAN runs in the 903 it would appear it was not supplied to owners of the older machines.

With the advent of the 905 it became more common for there to be an operating system and the final releases of a number of the Elliott systems moved to a command line-based user interface in place of using the machine control panel to jump to program entry points. (And while outside my own experience I would presume this facilitated integration with operating systems like RADOS).

There were other sources of software for the 903 beyond Elliotts: there was an active user group and members often made their software available to one another. Other languages such as BASIC and ML/1 became available through this route. Elliotts also had a CORAL system but it was not issued to 903 or 905 users.

Within Elliotts, other divisions at Borehamwood, Rochester and elsewhere developed their own variants of the standard software more appropriate to embedded use on 920B and 920M machines, and whose development systems

often lacked an online teleprinter. Examples include a more powerful editor (BOWDLER) and a two-pass SIR system that directly punched sum-checked binary tapes suitable for loading under initial instructions. Many of these utilities continued to support the older 920 telecode as well as the 903 and 900 telecodes, including provision for inputting in one and outputting in another. But this is a story for Terry Froggatt to tell as an Elliott insider.

## **Software Issues and Quality**

Elliotts labelled successive versions of each piece of software and documentation pages with issue numbers. These were incremented with each release but in a few cases there were minor releases with numbers like "Issue 4C". Most tapes were punched with a legible header giving the name of the tape and its issue number, although this was not done consistently and so some tapes lack headers, some have dates rather than issue numbers and some just have written or stamped legends. Surprisingly it was rare for the issue number to be included as a comment in source code or printed as a diagnostic when running binary tapes.

Users were expected to update the manuals with re-issued pages and to replace obsolete versions of software. Interestingly the software and manuals I obtained from Don Hunter seem to relate to a second 8K system he acquired from the British Ceramics Research Association and it would appear this organisation did a reasonable job of updating, although tiresomely Elliotts did not provide an updated index and catalogue against which users could check their holdings. The only way this can be done is to track through the release notes and other bulletins sent out by Elliotts. These are missing from Don's collection, but fortunately Richard Burwood has an almost complete set and has summarised the history of all the Elliott issued software in a helpful dossier. Terry Froggatt also has many of these notes, so with help from both of them I've been able to identify the last known releases for every item and ensure I have a "current" copy of both software and documentation.

As I've worked through all the software and documentation checking one against the other, mostly using my own 903 simulator as this offers a faster development cycle than paper tape on the physical machine, I've uncovered a surprising number of quality issues beyond the cavalier approach to tape labelling. Many of the examples in the documentation either contain significant errors and don't run, or don't produce the stated results. In part this is inevitable when documentation was prepared from a handwritten script by a typist, but some errors indicate nobody had ever checked out the examples. A case in point is QOUT1, a library routine for printing numbers, where the sample code

produces different results due to rounding errors than what was documented. Some of the later issues of the SIR mathematical routines contain simple syntactic errors, easily corrected, but again showing a lack of attention to detail. It's also evident from the sources that there was no common coding standard used by Elliott software developers, leading to irritating inconsistencies between how the same thing is done in different subsystems.

## Summary

The Elliott 903 came with three major programming systems: ALGOL, FORTRAN and SIR, together with associated utilities for software development using paper tape and mathematical libraries for engineering and scientific problems. There was little in the way of data processing facilities.

While one can quibble about some quality issues with how it was documented and distributed, the software generally did what was expected of it and made the machine a good system for teaching use and for use in scientific and engineering applications.

## Postscript

As a side effect of exploring Elliott software the author has written his own simulator in F#, a recently launched Functional Programming language developed by Microsoft Research, Cambridge. This simulator is based on the Froggatt/Hunter simulator (which was written in Ada) extended with a full implementation of the "undefined" effects of each model in the 900 range, interrupt handling, asynchronous input/output and an extensive range of commands for unpicking various paper tape formats, debugging and tracing etc. It has only been run on Windows 7 and is still a work in progress. Readers are welcome to request a copy. Included in the simulator are files containing all the Elliott issued software, demonstration scripts to illustrate their use and a manual documenting both the simulator and the Elliott software.

*A graduate of Leeds University, Andrew gained his PhD at the Cambridge University Computer Laboratory working with Roger Needham and Maurice Wilkes on the Cambridge CAP Computer and the Cambridge Ring. Following his time at Cambridge, Andrew founded his own company which made important contributions to middleware and distributed computing technologies. From 2001 until his retirement in 2011 he was the head of Microsoft's research lab in Cambridge. Andrew's email address is [andrew@herbertfamily.org.uk](mailto:andrew@herbertfamily.org.uk).*

---

## **Dekatron – The Story of an Invention**

*John & Robin Acton*

---

The publicity surrounding the relaunch of the Harwell Dekatron Computer prompted some of John Acton's old colleagues to press him to write the story of the Dekatron valve, which is the computer's key component. This was mainly written by John's son Robin, based on notes dictated by John, with a few later corrections and additions.

At the end of the Second World War a 24 year-old scientist called John Acton was working at the AC Cossor valve factory at Loudwater, Bucks. The transistor had not yet been invented, and the thermionic or "hot-cathode" valve was the active component of all radios, and all radar, military transmitters and receivers and the primitive attempts in computing and robotics. Valves were a vital part of the war effort – planes were once grounded because of the shortage of a valve called the VT60, and one valve at the Loudwater factory was being turned out at the rate of one every two seconds.

The hot cathode valves had a poor relation, the cold-cathode or glow discharge valve, which were filled with inert gases at low pressure using special pumps. Their main use was for stabilising voltages at some required level. Loudwater made several types in the "S1 Special Section", but they were always in trouble, with far too many failing to pass the test specification, leading to shortages and a terrible waste of precious materials and effort. Nobody at Cossor specialised in these devices, so at the end of 1944 John was appointed the quality control engineer in the S1 Special Section. He had considerable success at getting down the rejection rate, and soon became the expert in this obscure field of the valve industry.

When the war ended in 1945 most military contracts were cancelled and the workload eased, and John at last had opportunities to experiment. One of his experimental designs consisted of two cathodes, instead of the more usual single cathode, in a gas filled chamber with the normal single anode. The current was limited by the resistance in the anode line so that the bright cathode glow covered one cathode or the other, but not both. John proved that by sending a negative voltage spike pulse to the unlit cathode he could move the glow from one cathode to the other, and it would stay in the new position when the pulse ended.

Not only was this was the germ of a "two state device", but it had the added advantage that each state was signalled by the position of the luminous glow.

This was a new and purely electronic way of moving a light, in an era when moving light displays usually involved the use of mechanical relays.

At the time, John could think of no obvious use for this discovery, and it was not followed up. However in 1946 he was playing snooker with a friend who designed circuits for scaler devices. He told John about the state of the art scale of ten counter, which used four hot cathode double triode valves, some diodes, and ten lamps. "What we need is some bright bloke to invent a valve that has ten states built into it, instead of all this stuff". John thought about this problem and in 1947 conceived the idea of moving the cathode glow down a line of ten cathodes. He saw how this could be done, using intermediate "guide cathodes", but by now his job was designing a special cathode ray tube, so once again the idea remained just that – an idea.

In 1948 John was headhunted by Mr R.C. Bacon, the head of the new Ericsson Telephones valve design laboratory at Beeston, Nottingham and he was appointed chief valve designer. When he came to Beeston both John and Ray Bacon supposed the future lay in developing new hot cathode valves, and had no idea that cold cathode valves were the key to the future. However John was soon introduced to John Pollard, head of Ericsson's electronic laboratory, who showed him the standard mechanical 50 way uniselector device used to route telephone calls. John immediately realized that his glow moving valve could be developed to create a multiway electronic selector inside a gas filled valve.

The glow moving valve existed only in John's head, but he was at last able to draw sketches of his idea and ask the laboratory's highly skilled glass blower, Bill Revitt, to build a three-state glow moving valve from his drawing. There were to be nine vertical wires arranged in a circle of about an inch radius. Three of these were cathodes for stable states, and between each of these were guide cathodes which were to be pulsed in turn. While Bill assembled the valve to John's sketch, John built a simple circuit to produce the pulses to drive it from state to state. When Bill had got the device sealed in its glass envelope, John pumped and processed it and filled it with Helium, which he believed would give the longest life.

So early in November, just a month after he had come to the labs, and watched by Bill and two or three others in the lab, John was able to demonstrate the world's first multistate valve. As John pressed the button to send a pulse, the pale blue glow moved round from cathode to cathode. The idea worked! The news spread round the labs and then into the whole factory and people crowded in to see this strange new device. From that moment on John was always known in the factory as "John the counting valve man".

Bill then built another sample, with five states rather than three, which was demonstrated to Dr J.H. Mitchell, Ericsson's Head of Research. Dr Mitchell immediately grasped the potential value of the switching valve and authorized sufficient funds and resources to design and build a prototype 10-way version.

Things now moved with lightning speed. By Christmas, with the backing of the excellent tool room facilities, John had produced the design for a ten way counter valve that could be mass produced using conventional valve assembly and processing techniques. Called the GC10A, this was still using the Helium gas filling of the first experiments. In spite of its low counting speed (up to 400 pulses per second), such a demand was anticipated that a small production line was set up, soon to be superseded by a larger purpose-built valve factory.

Ericsson made protecting their rights an urgent priority, and a provisional patent (archived by Espacenet as GB19490001324) in the name of John Reginald Acton was granted on the 18<sup>th</sup> January, 1949 for "Improvements in or relating to electronic counting arrangements", the first of several patents concerning the new kind of valve. As was normal practice at the time, John as an employee of the company then assigned all rights in his invention to Ericsson. Patents



for the selector version soon followed, as well as patents in the USA. For these John had to execute a formal deed before a Notary Public, assigning his US patents to Ericsson in return for a payment of five shillings (25p)!

The valve as yet had no name. It was in a tea break that Ray Bacon suggested something like Decatron – "the ten device". Amended to the more correct

Dekatron this was name used when John's first technical data sheets were sent out to the potential users of the new device. The first article in the technical press using the name Dekatron, written by R.C. Bacon and John Pollard, appeared in *Electronic Engineering* in May 1950. The name was never trademarked, so although Ericsson held the patent rights, names like "Dekatron scaler" or "Dekatron timer" soon passed into the language, meaning based on the Dekatron valve.

It was very early in 1949, perhaps within a month of the first samples of the GC10A being available, that two famous scientists – Ted Cooke-Yarborough and K.K. Kandiah from the Harwell Nuclear Research Establishment – came to Ericsson to assess the possibilities of the Dekatron.

They were enthusiastic, and immediately found uses for the new valve. John always cherished his memories of meetings with these two friendly near-geniuses, and the confidence they showed in his ability. They seemed to sense that he knew his stuff and treated him as a friend as well as collaborator.

[Robin remembers Mr Kandiah coming to see his father at home, and joining the family's evening meal.]

The most spectacular project that grew out of this first meeting was the Harwell Dekatron computer conceived and built by Ted Cooke-Yarborough and Dick Barnes. In spite of its speed being laughable by today's standards the new computer replaced 20 manually operated calculating machines. Because of the low power consumption of the Dekatron (about a fifth of a watt), it was the first electronic computer to run off the mains, and needed no cooling fans.

The built-in reliability of its core component, the Dekatron valve that gave the computer its name, was a major factor that contributed to its success. John aimed to design the Dekatron to have a minimum working life of 10,000 hours, 10 times the accepted industry standard and 100 times the life of some of the cold cathode stabilisers he had looked after at Cossor. All the faults of the cold cathode valves he had wrestled with in the "Special Section" were carefully designed out, and the Dekatron and all the later valves designed by John had a working life of years rather than weeks. The GC10A was not just a revolutionary counter valve, but it set a new standard of reliability which was never surpassed.

Looking back, it seems a tremendous act of faith for Ted Cooke-Yarborough to have believed John's assurances that the Dekatron would have this reliability, for without this his whole computer project would be worthless. As the months passed John belief and Ted's trust were justified, and the Harwell computer with its banks of hundreds of Dekatrons ran almost continuously with a minimum of

maintenance for years. (In John's memory the early computer used the blue glowing GC10A, although he can't be certain about this.)

Ted Cooke-Yarborough showed that the way forward in computing was first to seek reliable operation and low power consumption, and then, and only then, to look for increased speed. The GC10A Dekatron was the first, and at that time the only, active electronic component to meet these requirements, so Ted used it and proved his point. The transistor, of course, also meets his criteria, and he was amongst the first to use it. But in 1949 that lay in the future.

A series of patents by John culminated in an improved design, filled with a neon/helium mixture instead of pure helium, which had an increased switching rate of 4000 pulses /second and a brighter yellow orange glow. As the GC10B, it became the standard Dekatron used by the thousand in scalars and timers, and even in primitive desk calculators and cash registers. The Dekatron scalars designed by K.K. Kandiah at Harwell for counting pulses from Geiger counters used 6 GC10Bs, coupled together to record units, tens, hundreds and so on and displaying the count by the position of the glow on a marked clock face.

The last Dekatron designed by John, the GC10D, extended the Dekatron principle so that the transfer was effected with single pulses, rather than the double pulse of the GC10B. Filled with a mixture of Helium and Hydrogen, it achieved a counting speed of up to 40,000 pulses per second, 100 times as fast as the original GC10A.

But by now primitive transistors were available and the glory days of the Dekatron were coming to an end. John left the Ericsson Valve Division in 1959 to follow new lines of research, and by the mid 1960s the Dekatron was effectively obsolete and production at Ericsson had ceased. It had, however, been copied by valve makers all over Europe, the USA and Russia, and some companies went on making their versions. Right up to the end of the century Dekatron timers were still in use for spot welders, with the original Ericsson valves.

A web search for "Dekatron" brings up a wealth of information about the Harwell Computer and rather little about the valve itself. However, [tinyurl.com/dekatronpics](http://tinyurl.com/dekatronpics) shows pictures of Dekatrons in many applications. [tinyurl.com/Dekdatasheets](http://tinyurl.com/Dekdatasheets) shows scans of some of the data sheets written by John. [tinyurl.com/GC109Bs](http://tinyurl.com/GC109Bs) has some excellent photos of Ericsson GC109Bs.

---

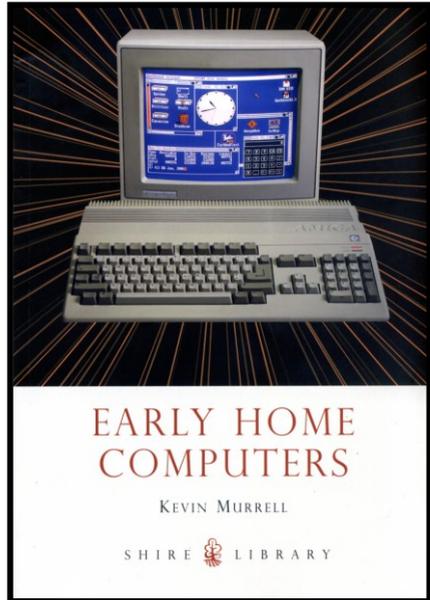
# Book Review: Early Home Computers

*Dik Leatherdale*

---

It is perhaps reasonable to suggest that this new book by Kevin Murrell, CCS's energetic secretary, falls outside the normal comfort zone of *Resurrection* readers. None the worse for that though.

At first sight, given the slim size of the book, the identity of the publisher and the sheer number of high quality illustrations, many taken from contemporary advertisements, one might be forgiven for concluding that this is a rather lightweight tome. But that would be to underestimate it, for it contains a narrative which is obviously the product of much research. And if "Ooh, I didn't know that!" strikes the informed reader as often as it did to me then it's obviously not been time wasted.



The themed chapters are in broadly chronological sequence, beginning in the 1970s with the predecessor systems – minicomputers, progressing through the so-called *homebrew* or hobbyist phase and then plunging into commercially-produced systems for the home, not least the Sinclair ZX machines and the BBC Micro. All this seems to have proceeded alongside the serious business of producing the "proper computers" with which most of us were once concerned without much contact between the two groups. IBM and Apple brought the two strands together. Our story ends in the early 1990s with the adoption of the CD and the modem.

I started this review by suggesting that this isn't quite our usual fare. I have found that whenever I talk to younger (i.e. middle-aged) ex-colleagues about my involvement with the CCS, they all say "You mean ZX81s and BBC Micros. How interesting!". Well no I don't. But their reaction suggests that there may be a ready audience for this book outside of our own circle as well as within it.

---

## Forthcoming Events

---

### London Seminar Programme

21 Mar 2013	The Atlas Story: 1956 to 1976	Simon Lavington
18 Apr 2013	CAP – The Last Cambridge Computer	Andrew Herbert
16 May 2013	Babbage’s Analytical Engine	Doron Swade

London meetings normally 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](mailto:r.johnson@bcs.org.uk), or by post to Roger at Birkbeck College, Malet Street, London WC1E 7HX.

### Manchester Seminar Programme

19 Mar 2013	Computing Before Computers — From Counting Board to Slide Rule	David Eglin
-------------	--	-------------

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](mailto: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](http://www.computerconservationsociety.org/lecture.htm). Details are also published at in the events calendar at [www.bcs.org](http://www.bcs.org).

### CCS Website Information

The Society has its own website, which is located at [ccs.bcs.org](http://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://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.

## 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](http://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](http://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](http://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](http://www.sciencemuseum.org.uk) for more details.

**Other Museums** : At [www.computerconservationsociety.org/museums.htm](http://www.computerconservationsociety.org/museums.htm) can be found brief descriptions of various UK computing museums which may be of interest to members.

### Contact details

Readers wishing to contact the Editor may do so by email to [dik@leatherdale.net](mailto: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](mailto:kevin.murrell@tnmoc.org), or by post to 25 Comet Close, Ash Vale, Aldershot, Hants GU12 5SG.

### North West Group contact details

Chairman	Tom Hinchliffe:	Tel: 01663 765040.
Email:	<a href="mailto:tah25@btinternet.com">tah25@btinternet.com</a>	
Secretary	Gordon Adshead	Tel: 01625 549770.
Email:	<a href="mailto:gordon@adshead.com">gordon@adshead.com</a>	

---

## 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:**

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: **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

Bletchley Park Trust: **Kelsey Griffin:**

kgriffin@bletchleypark.org.uk

TNMoC: **Dr David Hartley FBCS CEng:**

david.hartley@clare.cam.ac.uk

### 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:**

ccs@tjf.org.uk

Ferranti Pegasus: **Len Hewitt MBCS:**

leonard.hewitt@ntlworld.com

Software Conservation: **Dr Dave Holdsworth CEng Hon FBCS:**

ecdh@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

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 FBCS:**

doron.swade@blueyonder.co.uk

EDSAC: **Dr Andrew Herbert OBE FEng FBCS:**

andrew@herbertfamily.org.uk

Tony Sale Award: **Peta Walmisley:**

peta@pwcepis.demon.co.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

## Point of Contact

Readers who have general queries to put to the Society should address them to the Secretary (see page 36 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.

Editor — Dik Leatherdale

Printed by — BCS the Chartered Institute for IT

© Computer Conservation Society