Computer Conservation Society

Aims and objectives

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

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

The aims of the CCS are to

◇ Promote the conservation of historic computers and to identify existing computers which may need to be archived in the future

◇ Develop awareness of the importance of historic computers

◇ Encourage research on historic computers and their impact on society

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

The CCS is funded and supported by a grant from the BCS, fees from corporate membership, donations, and by the free use of Science Museum facilities. Membership is free but some charges may be made for publications and attendance at seminars and conferences.

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

The corporate members who are supporting the Society are Bull HN Information Systems, Digital Equipment, ICL, Unisys and Vaughan Systems.

Resurrection

The Bulletin of the Computer Conservation Society

ISSN 0958 - 7403

Number 14

Winter 1995/1996

Contents

Editorial
Nicholas Enticknap, Editor **2**

News Round-Up **3**

Piece of Babbage History Sold
Doron Swade **4**

Deuce — its life and times
Jeremy Walker **5**

Telephone Traffic and Other Hobbies
Don Hunter **15**

Experiences with Pegasus 1
Donald Kershaw **22**

Letters to the Editor **30**

Forthcoming Events **32**

Editorial

Nicholas Enticknap, Editor

Progress is continuing in the development of the Bletchley Park Trust Museum, and visitors who take advantage of the guided tours (see Forthcoming Events on page 32 for details) have plenty to see. Our Secretary, Tony Sale, reports that the project to build a replica Colossus code-breaking computer in particular is coming along nicely.

As yet, however, we still await the completion of the sale of the Park to the Trust. Another disappointment is that the application for a contribution to the Museum from the Heritage Fund, out of the proceeds of National Lottery money, was initially turned down. This decision is now under review, and Society members can help the cause by writing to Heritage Secretary Virginia Bottomley at the House of Commons urging her to decide in favour of the Trust.

The feature content of this issue covers experiences with three early British computers. For the first time, we carry a piece devoted to the English Electric Deuce system, based on a talk given by Jeremy Walker to the North West Group.

Don Hunter has contributed a piece on the Zebra, which supplements his article in issue 11 by outlining some of the STC prehistory and by discussing some Zebra applications. This article is based on Don's talk at the Zebra seminar in London in October: we plan to publish other articles from this event in future editions. Articles based on the all day Leo seminar in November are also in preparation.

Our final main feature is based on another North West Group talk, and describes Donald Kershaw's experiences as a user of Pegasus. Enthusiasts for this machine will also want to turn to the Letters section, where Derek Milledge has arrived at the definitive explanation for the mysteries of Pegasus package numbering.

Many readers will have seen news items on television and in the national press last October about the sale of part of the original Difference Engine. Doron Swade gives the details behind this story on page 4.

Finally, we are now in a position to provide the whole text of *Resurrection* in electronic form via the Internet — not just the current issue, but all the previous 13 as well. Full details of how to take advantage of this can be found on the facing page.

News Round-Up

The Society now has an electronic archive, from where members with access to the Internet and who can use FTP (File Transfer Protocol) may download files. The archive will be updated regularly.

At the moment there are two main sections. One contains every issue of *Resurrection*, in three formats. These are LaTeX (which we may not continue to support, and which lacks the first three issues), Microsoft Word for Windows, and plain ASCII. Illustrations are provided in TIFF (Tagged Image File Format) or GIF (Graphics Interchange Format), and are downloaded with the issue's text file.

The second section is devoted to simulators, some of which have been provided by members while others are publicly available. There are simulators for the Manchester University Small-Scale Experimental Machine, Edsac, Stantec Zebra and Pegasus. Any member with another simulator that they would like to add to this collection should contact Chris Burton as soon as possible, so that we have a really comprehensive collection. The simulator should ideally be accompanied by well-written documentation of the target historic machine and instructions for operating the simulation software. Chris is happy to advise on both requirements.

To access the archive, connect to **ftp.cs.man.ac.uk** as an anonymous user, and change to directory **pub/CCS-Archive**. The directory structure from there is straightforward, and there are ReadMe files to provide guidance. We are indebted to Professor Frank Sumner and the staff at the Manchester Computer Centre for hosting our archive site, and to member David Mitchell for arranging the access details.

Wanted

We are trying to locate the following articles for Society projects.

- Five 19" Post Office equipment racks, at least 6' 6" high — the sort with side members made of rolled steel channel 3" x $1\frac{1}{2}$" and drilled and tapped all the way up both front and back.

- Scores or even hundreds of tubular capacitors — the sort that come in a metal can, possibly covered with a clear plastic film sleeve, particularly 0.5 and 0.1 microfarad, 350 V working. The 0.5 microfarad units were typically $\frac{3}{4}$" diameter and $2\frac{1}{2}$" long.

Anyone who can help should contact Chris Burton.

Piece of Babbage History Sold

Doron Swade

A small demonstration piece of Charles Babbage's famed Difference Engine No 1 was auctioned at Christie's on 4 October 1995. The piece was tentatively valued at £50,000 prior to the sale. In the event bidding was fierce and foreign interest strong. The hammer price was £160,000 and the piece went to the Powerhouse Museum, Sydney, Australia.

The device is one of several similar demonstration pieces assembled in the late 1870s by Henry Prevost Babbage, Charles Babbage's son, after his father's death in 1871. Henry Prevost, who took a strong interest in his father's work, and to whom Charles bequeathed his workshop and drawings, put together about six small assemblies from unused parts intended for the great engine, the construction of which was abandoned in 1833. These he sent to centres of excellence to draw attention to his father's work. Examples went to Cambridge, University College London (this piece is now at the Science Museum, South Kensington), Manchester, and Harvard.

The piece auctioned had been in the Babbage family for over 100 years and was, until then, in the possession of Jean Babbage in Auckland, New Zealand. Having travelled from Auckland to Christie's the itinerant device has now returned to Australasia. The Science Museum did not bid as the piece duplicates one already in its collections and on display in South Kensington.

The assembly is an adding device with carry mechanism. In operation it demonstrates the basic logical element of Difference Engine No 1 which is repeated many times in the full machine. The parts were made in the workshop of Joseph Clement, Babbage's engineer, and were completed prior to 1833. Most of the 12,000 completed parts for the engine were later melted down for scrap. The demonstration models assembled by Henry Prevost, and the large assembly in the Science Museum consisting of 2,000 parts, are the only substantial mechanical assemblies to survive.

Doron Swade is Senior Curator (Computing and Information Technology), Science Museum, South Kensington.

Deuce — its life and times

Jeremy Walker

The Deuce team at Kidsgrove took a design proven on prototypes and made it fit for routine production and reliable operation. They also undertook the ongoing development of the system, adding further peripherals, high speed store and computing functions. This is an account of one of the most seminal and undersung pioneering computing initiatives.

English Electric didn't seem to get the publicity we think we should have had. Whenever I see something reported about early machines, Deuce is ignored. Readers who have read the Official History of ICL will have found it interesting, but I wish that the author had looked more into what English Electric achieved.

In retrospect, this lack of recognition was probably for two reasons. First, English Electric had decided that, as a big engineering company itself, it needed computers, and people at the Royal Aircraft Establishment (RAE) Farnborough were quite certain that they also needed this new tool. So English Electric's first objective was to use Deuce and to sell it to a small number of specific customers: there was no need to market the machine, nor to seek publicity.

Secondly, Ferranti was a lot more commercially minded, and did a very much better job of marketing and publicising their activities.

Deuce takes its place in a continuum of computing developments. From the National Physical Laboratory (NPL) came the Pilot Ace. This was intended to be a prototype of the Ace (Automatic Computing Engine), which we then engineered as the Deuce. Deuce itself was developed through three stages: the Mark 1, 2 and 2a.

At the same time, English Electric Computers (EEC) developed (independently of Deuce) a process control machine called KDN2. It realised that with a little tarting up it could become a commercial machine, and developed it into KDF6, and later KDF7.

KDF9, that well-known 'stack machine', was also developed at Kidsgrove. It was technologically independent of the other machines, and found favour with computing-intensive customers. We then got together with RCA and re-engineered the RCA501 to become the KDP10 (later the KDF8), aimed entirely at the emerging commercial marketplace.

The story starts just after the War, when Alan Turing returned from Bletchley Park to NPL with a "stack of documents which effectively was the specification for Ace". So the first project at NPL was to be a pilot Ace, run by Harry Huskey (of the US National Bureau of Standards) and Sir Charles Darwen (of NPL). They put together the necessary development team, starting with three people charged with setting up an Electronics Section to build it: Messrs Wilkinson, Davies and Newman.

Sir George Nelson (later Lord Nelson of Stafford), Chairman of EEC and at the time on NPL's Advisory Council, offered English Electric's help in this endeavour, and Colin Haley was recruited by the company in 1949 to make it happen. It was he who coined the name Deuce — both an acronym for Digital Electronic Universal Calculating Engine and the logical successor to Ace.

Colin still today describes the Ace pilot as a "dog's breakfast", and from 1949 through to 1951-52 English Electric set about engineering what was very much a laboratory model into a more robust entity, as Deuce.

Wilf Scott, who later became Managing Director of English Electric Computers, together with RAE Farnborough, decided that the two companies had a real requirement to deploy this computing power in their businesses. So seven Deuces were built at the Nelson Research Laboratories (NRL) at Blackheath, with first delivery in 1952. Some time after that, it was realised that Deuce was no longer a laboratory curiosity, and should be sold commercially by EEC.

To this end, continued development and readying for production was transferred in 1954 to the recently formed Industrial Electronics Department (IED) at Kidsgrove. A little later, in 1955, I joined IED from English Electric at Bradford when my new boss, Derek Royle, was given the job of recruiting a team to take the machine forward.

Despite the re-engineering of the product at NRL, it was hardly reliable and so a good deal of work was to be done on making it so, and on enhancing it as necessary. Kidsgrove became responsible for all further development and output.

Our customers came with very few exceptions from the scientific community. Farnborough was interested in airframe stability and flutter, English Electric's Atomic Energy Division at Whetstone was interested in Monte Carlo methods of neutron capture prediction and statistics, and BP in Aldgate, London analysed seismic studies for oil exploration.

There were a number of multiple sites — there were three at NRL but that was still very much in a laboratory environment. More relevant multiple sites were Farnborough where there were two and Bristol Aeroplane Company (now BAe) at Bristol. MAFF (Ministry of Agriculture, Fisheries and Food, at Guildford) was unique in having three machines which were, as I understand it, entirely used for commercial data processing.

The initial configuration of the two Deuces at Farnborough included only 32-column card input/output (I/O). We used to refer to them fondly (though fondly was perhaps the wrong word in view of the trouble they gave us!) as Hollerith machines, but they were actually supplied by BTM (British Tabulating Machine Company). They were the Balancing Tabulator (which read cards at 200 per minute) and the Gang Punch (which output at 100 cards a minute).

For memory, we had 400 32-bit words of high speed store in ultrasonic mercury delay lines and 8192 words on magnetic drum. I became very expert on the drum system, as, for a year or so, the servos which controlled the drum's phase-locked rotation and head-positioning were, to put it mildly, somewhat marginal in their operation.

Deuce had monitors in which one could look at the contents of the mercury store and see everything happening as computing proceeded. The absence of such a facility on subsequent machines came as a horrible shock to me. I believe you could do the same thing on the Ferranti Mark 1 or Mark 1* and, for all I know on the Pegasus, but we haven't been able to do it since!

Some of the people involved in the development of the machine were Turing, Colin Haley, Cliff Robinson, George Davis, Derek Royle and of course myself. Some of them are no longer with us but most are alive and well. Others who played a significant role included Jack Richardson, John Boothroyd, John Newman, Derek Savoury (located until relatively recently at ICL Bracknell) and Vic Matthews.

Deuce possessed two noteworthy attributes. One was that because acoustic delay lines are serial devices and inherently slow, a method of structuring the instruction word was chosen so as to enable one to program optimally the order of instructions. As a result, if you did it correctly, you could actually pick up instructions in sequential minor cycles, which were 32 microsecond timeframes. This, it was believed, would overcome the problems of the slowness of access to the lines.

The other attribute was called "simultaneity". It really came big in the 1960s or 1970s, but in the 1950s Deuce was able to operate all its

I/O independently of instruction processing, whether it was card or drum or later from paper tape and magnetic tape. All those operations could proceed together, and we made good use of this attribute in two particular areas.

First, although initially only binary data could be read from the 80 column cards, there was enough time between reading any two card rows to be able to carry out binary-to-decimal conversion. Similarly, in the multiplier-divider unit, it was possible to calculate the sign and do any rounding that you wanted to do (neither of those operations being automatic) while the autonomous multiplier-divider unit was actually operating.

I come on to the specification of the machine, at least of the Mark 1 — we subsequently added more delay lines, magnetic tape and so on. Deuce basically was a serial machine, little different to any other of its type. The so-called *long* multiplication and division took two milliseconds.

The mercury delay lines were housed in a thermostatically-controlled enclosure known, from its appearance, as the 'Mushroom'. It was temperature controlled at 50 degrees Celsius and a source of worry from time to time, as the line length and thus the apparent data capacity of the storage lines was very susceptible to temperature variation.

We also had a number of short delay lines — four Single Word, three Double Word and two Quadruple Word — which gave much faster access to the data (the Single Word line gave you that single word more or less instantaneously whereas if you wanted one word out of the 32-words long lines, you had to wait for the other words to go past). A Quad Line was about 600mm long, including the circuitry (common to all lines) necessary to get data in and out.

The idea behind a Serial Store is that it is necessary to pick the right moment at which to access or interrupt the pulse sequence, in order either to insert new data or to extract the data that you want. So access was rigorously sequenced and clocked by the Control Logic, and was critically dependent on the maintenance of an exact physical length of the path length in the mercury.

The most common circuit in the machine was the traditional long-tail pair. It was a bi-stable device which was fitted with a large cathode-feedback resistor R7 which effectively stabilised the current through the valve. Because current was switched either to flow in the lefthand or the righthand half of the valve, the effect was to provide a more constant load on the power supply so that it did not need to be regulated.

Within the drum system, the actual rotating drum was about 150mm in height. It had 256 tracks and rotated at about 6500 rpm. Of the two servos, one controlled the rotational speed and phase of the drum, locking the phase of information recorded on it to the basic 1Mhz clock, with position-sensing from a very precisely cut toothed wheel with 1024 indentations on the top of the drum.

At Farnborough, just the other side of the canal, there was a company called Pyestock which used to test big turbines. On Start-up, the associated motor starting current used to depress the mains voltage over the entire area, the drum dropped out of synchronisation, and you'd had it! Because phase position had been lost, your data couldn't be recovered.

The other servo was the Head Shift Servo. There were two sticks each of 16 heads, one on each side of the drum. They were driven by moving coils in magnets, with feedback of vertical position by contacts on two arms sliding up and down a linear potentiometer, called the 'Reset Pot'.

The way that position feedback was obtained was by applying a voltage across the potentiometer, such that the sliding-contact connected to the head-stick would pick off a voltage representing the position of the stick, and thus the heads.

Initially the tip of this arm was just a copper blade. Unfortunately, the knife edge of the copper blade fairly quickly wore and therefore the position was indeterminate. Also, the wear debris shorted out the turns of resistance wire on the pot, so making the positional feedback voltage non-linear. We eventually found a palladium alloy called JM77 from Johnson Matthey which overcame this problem.

The drum had a perspex cover, necessary because although the machine, in theory, needed no forced cooling, it actually emitted some 79 kilowatts of heat, which made it rather uncomfortable for people in the room. So the machine itself was cooled by pumping in lots of air — but you know what comes in with air. However well filtered, over a period of time dirt accumulates. It wasn't long before we were discovering fine lines on the drum — dust was getting in between the heads and the oxide surface. So the drum was put inside its own little enclosure, separately blown through a filter.

The power supply unit (PSU) is worthy of further mention — not least because it was very nearly the size of the machine itself! It incorporated, *inter alia*, six big power supplies which produced +/− 100, 200 and 300 volts, as well as lots of amps. If you wanted to examine circuitry, the best way of doing it was to get hold of the handle of the chassis and stand on the baseplate. Unfortunately, on the chassis it was possible for your fingers to touch the terminals, and you could quite easily find yourself across +/−300 volts. This made one jump, to put it mildly!

Although the PSU was not stabilised, we did have, in places where there was long-term drift in the mains voltage, a mechanical voltage-sensing device and a big auto transformer about one and a half metres tall that kept the mains within its normal bounds.

The initial machine used some 1400-odd valves of nine types, of which the most common was the ECC91 common-cathode double-triode. Another was the EL81, which was used to drive the head movement on the drums. The failure rate of valves was typically about 40,000 hours, so with 1400-odd valves in Deuce it doesn't take a genius to work out that they failed fairly frequently.

In those early days, maintenance and fault-finding, though disciplined, was very much a case of "Hello, I think I'm on to something here". Marginal checking on Deuce was the order of the day, recognising that valve characteristics were very prone to drift.

Checking was done with a Bias Box. One could go round the chassis and insert a plug, driven by a cathode-follower, into any of the valve stages. A potentiometer in the box could change that valve's grid level up or down, to show how safe was the operation of that stage. A resistor value was changed if it wasn't: there were a lot of stages to check but they were fairly stable and were all checked on a monthly basis.

There was also a key on the Control Panel which permitted one simultaneously to alter the bias points in the entire machine up or down, positive or negative, and thus see whether or not an operating program was going to fail in the extreme positions, so providing a margin of safety in the normal position.

We discovered, quite late on, that an ordinary battery was a good way of carrying out this biasing, and we tended to use them in order to avoid an unfortunate glitch which could take place when one plugged in the Bias Box. There could be enough of a surge actually to cause a test program to fail and, if one was actually trying to establish whether it failed because of marginal drift, this wasn't useful.

Eventually the result of this freedom was to have half a dozen battery units. Frequently, visiting a site, you would find all the doors open and see these things plugged in all over the place. This was usually because someone had to get their program through the machine, and offsetting biasses in this way was the only quick fix — you needed a lot of time-out to achieve a permanent repair.

The machine had no parity, no other automatic checking and no automatic error recovery, though in the later additions of paper and magnetic tape, parity was included in those I/O circuits.

Parity was not included in the drum system because the track length was 1024 bits, and the argument was that to have a single parity bit in 1024 isn't terribly helpful. As a consequence, all sorts of clever programs were put together which used check summing to ensure any errors were highlighted.

Towards the end of the production programme a form of automatic 'double-read-and-compare' was introduced in the drum circuitry. This was a tremendous benefit: if the Compare failed then a (program-controlled) re-read was forced.

The Maintenance Period was two hours a day. I used to advise our maintenance engineers round the country, "Do not be talked out of your two hours just because the machine was faulty yesterday and your customer lost time: surely it needs more maintenance not less".

We used to age the valves (because of that 40,000 hour failure rate). We had a little test rig in which we used to operate valves for 100 hours, rejecting those that failed — drifted — prematurely.

That brings me to vibration testing. At the time we used to refer to some intermittent faults as "tappy" faults, and we would beat the machine to death in trying to reproduce a failing condition.

Anne Wesson was a maths postgraduate student at Glasgow University when we delivered a machine there, and she was appointed by Dr Gillis, manager of the new Computer Centre, to adjudicate on the Customer Acceptance Test. The Acceptance Test at that time required three days of totally fault-free running — remember, there was no automatic error-recovery and precious little diagnostic ability.

We learned in the end that the only way of getting these machines through such tests was to subject every valve to a test for "microphony" (very heavy vibration). In other words, we hit it! Many's the time that we banged too hard and shattered the glass. The blow also shorted out the internal electrodes, and that took the entire machine off with a bang and a crash. However, it was the only way you could get it to run for about four days. The valves started to become microphonic again in due course — this was perhaps the worst problem in normal operation.

We formed an Expert Help unit, the Digital Computer Mobile Service Unit, which was later divided into DCMSU/North and DCMSU/South. Jack Richardson, who sadly was killed in an air crash about 15 years ago, and I formed this unit in about 1958. We had a van packed with oscilloscopes, signal generators and spare parts, in which we used to dash up and down the country.

Development was constant. Deuce Mark 1 started off with only 32 column binary input and output by cards, but was fairly quickly changed to 64 column (still binary). We then attached the IBM 528 Accumulating Reproducer, modifying it that so we could have 80 column decimal input/output. Later still, we added further high speed storage in the form of another Mushroom full of delay lines. We also added the Automatic Instruction Modifier, and a paper tape reader and punch.

The most common fault, funnily enough, wasn't dry joints; they plagued us enormously on KDP10, which gave us our early experience of printed circuit boards. No, the problem on Deuce was unsoldered joints. The production people at Kidsgrove inevitably missed a wire-joint somewhere.

Another fault with which we had a lot of trouble initially was that all the power lines to the individual circuits went through feed-through decoupling capacitors. These used to fail spectacularly, going off with a bang from time to time.

The drum servo synchronism would go out of sync whenever we had really severe mains spikes. This could be extremely annoying if it happened in the last hour of a three day Acceptance Test, because it was by definition a failure.

I mentioned that the line enclosure was stabilised at a temperature of 50 degrees C and the precautions taken in its design to mitigate against draughts. That was fine unless it was in a *strong* draught. We had one or two occasions where people would come in and leave doors open — it wasn't proof against that!

When the delay lines were first put together by our colleagues at NRL, the crystal holders immersed in the mercury were made of perspex. Perspex has a high thermal coefficient of expansion so, as the thing got hot, and even within the limits of the temperature controller, the air gap between the electrode driving the crystal and the crystal itself would vary so that the line length became wrong. We subsequently changed to an Araldite crystal holder.

Card reader brushes were a problem, and card jams. Hours and hours were spent trying to prove that one particular colour of card — you may remember that they had colour stripes — was the cause of the trouble, somehow affecting the stiffness of the card. I remember at Farnborough we spent days convincing ourselves that grey stripe cards were the reason for our card jams. This turned out to be *true*! The three boxes of grey stripe cards in the card store were close to a leaking radiator!

I shall finish this article with some 'funnies', hoping that the humorous will stick in your minds.

Deuce had a door at the back of the cabinet frame, allowing access to change the valves or whatever. Inside it was nice and warm and fairly private. I did some of my courting inside these machines: also, on one occasion after I had married, I went late one night to fix a machine at Warton and took my wife with me. The machine room was cold, but it was warm inside the computer so she took a book and a chair inside!

One favourite trick played by those of us who smoked was, when somebody was peering into the monitors trying to work out what had happened to one of his programs, to go inside and puff cigarette smoke into the back of the Control Panel. Smoke would issue from the front round the sides, and more than once someone shot across the room to the Power Supply Unit to turn off the machine before it burst into flame!

I've mentioned the 600 volt chassis-shock: that was a very real problem, though no-one was ever actually killed by the shock. However we had some informal training for the experience. People used to charge up capacitors to this voltage and throw them, without warning, to someone. If, catching it, you happened to hit the terminals you had yourself a nice little shock!

Another problem with mercury is that it is fairly easy to spill when you are trying to mend a delay line. Many's the time I've chased blobs of mercury around the inside of the Mushroom with a straw with flux on its end — gooey flux — because (and not many people know this) you can pick up mercury with sticky flux.

It was an aluminium enclosure and the stories at the time were that if you left loose mercury in there it would amalgamate with the aluminium. I don't know whether that was true but when you look back and think of mercury vapour and the Health and Safety at Work Act... I used to chew pieces of PVC-covered wire too, and there are carcinogens in PVC!

Another funny — though those of us to whom it happened weren't too pleased — related to the drum mechanism. It took something over 20 milliseconds to change head positions on the drum, so it took that length of time between initiating a requirement for data off the drum and the time you got it. If you tried to access the buffer store during this period you would get garbage.

Therefore a valve stage known as the Control Magnetics Interlock (CMI) provided a 35ms interlock. In order to carry out a check of the head-shift time, one removed the CMI valve in order to disable the Interlock. It was easy to forget to put it back, so when you handed the machine over to the customer saying, “It’s alright now, it’s shifting heads within 25 milliseconds, plenty of safety”, it didn’t work. One could spend hours looking for a fault... the fault was that the CMI valve was out!

At one stage ITV decided to use Deuce for predicting Election results. Since the machines were far from portable, ITV had to come to us. I remember one occasion in Stafford and another in Kidsgrove.

Around then, small transistor radios were just becoming available. I had made one from a Sinclair kit. I was using it in the computer room, and it was obvious that it was picking up electromagnetic interference from the machine. The guy in charge of feeding the data back to ITV was intrigued with this, so that we positioned the radio inside the machine — for better reception — and placed a microphone so that as the programme was running, this “radiophonic” noise was broadcast live. This has to be the earliest example of electromagnetic incompatibility.

Editor’s note: this is an edited and abbreviated version of a talk given by the author to the North West Group of the Computer Conservation Society, at The Museum of Science and Technology in Manchester on 7 February 1995. Mr Walker wishes to thank Colin Haley and George Davies for assistance in historical research and Jenny Wetton, Curator of Science, Museum of Science and Industry in Manchester and Jackie Bull, Secretary, for the initial transcription from audio-tape.

Telephone Traffic and Other Hobbies

Don Hunter

This article describes the author's involvement in the design of two STC projects, Step I and Strad, which became important influences on the design of the Zebra computer. It also discusses the application which provided him with his major user experience of the Zebra — the problems of calculating the effects of telephone traffic on exchange efficiency.

When STC set up STL (Standard Telecommunication Laboratories) at Enfield in 1946 one of the staff members was Alec Reeves, the inventor of Pulse Code Modulation. STL was 20 years further on responsible for another major advance, when Charles Kao and George Hockham published a landmark paper on optical fibres.

I started work there in 1951. In those days we clocked on and off; some of us were paid in cash on a Wednesday for the previous week's work; and there were no job numbers.

The stores contained an amazing collection of both electrical and mechanical components, yet someone still had to go once a week to Lisle Street to buy obscure items. Once he bought me some 4.7 ohm resistors for an ingenious ferrite-core shift register, but as the wireman was putting it together I found that they should have been −4.7 ohms instead. That's the trouble with development: so many things don't come out right.

The staff in the workshops were able to make waveguides for the radio people, and valves too — things like klystrons. One day I had to shift a nut on a magnetic drum and went in to borrow a large spanner, but they would not give it to me. I returned at lunchtime pretending it was for my motor bike, and that was all right!

As a member of the ITT group, we had some collaboration with other ITT companies, particularly BTM (Bell Telephone Manufacturing Company) in Antwerp and SEL (Standard Elektrik Lorenz) in Stuttgart. I spent six weeks at BTM on a prototype electronic exchange, several months in New York and New Jersey, and a year at STL's sister lab, LCT (Laboratoire Central de Télécommunications), in Paris.

Our Zebra computer was installed in 1960, a year or two after STL moved to Harlow. There were two significant projects which helped to smooth the path for its arrival. The first was a computer called Step I (the initials stood for Standard Telephones Electronic Processor). I am grateful to Bob Newton for providing information about it. The second project was Strad, a store-and-forward message switching system.

The Step I story starts when STL was asked by STC to design a computer in early 1954, for use at Woolwich mainly for filter calculations. Joe Rice, Peter Harrild and I were expected to complete the logical design within four months.

We chose a magnetic drum as the main store and gave it an order code exactly like Edsac I with a few extra instructions, such as negative accumulative multiplication. We adopted microprogramming for its control from Edsac II, using a diode matrix to implement the microcode. Of the 64 micro-orders, 35 were used initially.

There were six internal registers using pairs of heads on the drum, but none was available to the user and there was no index register. It was a two-address machine with a memory capacity for 2048 instructions of 28 bits, although only 512 were ever provided because an existing experimental drum was this size and the development of a larger drum was considered to be too expensive. The clear plastic moulded cover for the drum was in great demand as a butter dish!

The drum ran at 3000 rpm and provided a 25 Kcs clock speed. There were only 16 words on each of the 32 tracks, which were selected by junction transistors. There was always at least one faulty track, and there were no spares. This lesson was well learnt, and the Zebra drum was provided with spare tracks.

The circuit design and construction were done in STC Newport by an enthusiastic team at the Information Processing Division led by Fred Filby. When finished in early 1956 it ran to 10 bays (8' high x 2' wide x 2' deep) and probably more than 1000 valves, with the individual units soldered in place. After the experience of changing units during the course of normal maintenance, STC chose to construct Zebra with plug-in units.

I was sorry never to have seen Step I in operation, because with the covers off one could watch the neon lights on the rows and columns of the microprogram switch.

As an example of the flexibility provided by a microprogram, a hardware divide operation was added after discovering that Step I took 17 seconds to perform a programmed division. Was it really that slow? There was a clerihew which went:

"The computer in South Wales

Does the work of several snails
By working hard day and night
It gets a long division right."

Step I ran until 1959 at STC Woolwich doing mainly filter calculations, and formed a useful warming up exercise for Zebra.

The other STC development was Strad. In the late 1950s STC built a store-and- forward message switching system using a 10 inch diameter, 10 inch long drum for storage — at last a drum of respectable size! The control was entirely in hard-wired logic.

The equipment filled a large room at Gatwick airport. I remember paying a visit there to investigate electrical noise problems and thinking that one might replace a printed circuit card in the wrong place, because you could easily wander into the wrong avenue of bays.

Telephone Traffic

When you make a telephone call and hear a "number unobtainable" tone, you may be puzzled, and perhaps conclude that you dialled a wrong number. The culprit is more probably internal congestion in the telephone exchanges between you and your called number. It cannot be prevented — at least not without making exchanges quite unreasonably larger.

What is the chance of losing a call in this way? Probably 1% of calls are lost in each exchange on average: the percentage varies greatly according to the number of calls being handled — the so-called traffic.

Inside an exchange are switches where calls come in on one set of circuits and are offered to other circuits which connect in turn to other switches. There are lots of ways to interconnect them — this is known as the trunking, which is just the architecture of an exchange. The designer needs to know how well the traffic will flow through the whole exhange.

Let us look at this problem for a moment. Say 12 calls on average are being offered to 15 circuits: the problem is to calculate the chance of all the circuits being engaged when a new call appears. (An identical problem is if 12 people are typically complaining at any one time to an organisation which has thoughtfully provided 15 people to deal with them.) There are three ways of doing this.

The first is to calculate the outcome using Erlang's loss formula, named after the Danish mathematician AK Erlang. A unit of traffic is called an erlang, so we have 12 erlangs flowing in this case.

The second method is by simulation. The idea of using computers as simulators of telephone traffic originated with Kosten who gave a talk entitled “Fictitious Traffic Machines” at a conference at the Cambridge Mathematical Laboratory in June 1949. Then there was some pioneering work by Neovius and Wallstrom at LM Ericsson in Sweden in the mid 1950s.

For the case of a erlangs offered to n circuits, you have a source of random integers and a rule that you try to release a call if the integer is in the range 1 to n, and you attempt to make a call if the integer is in the range $n+1$ to $n+a$. This mimics the case of calls where the holding time is exponentially distributed, and in fact we are lucky that we do not have to remember the holding time during the simulation.

The third method is the use of equations of state. That is, you consider the probabilities of having 0 calls, 1 call, 2 calls up to 15 calls. There are 16 states corresponding to having any number between zero and 15 calls. There are also 16 equations which relate the probability of being in a particular state to that of its neighbouring states, and the coefficients can be found from the offered traffic.

But these equations are not an independent set, so they have many equivalent solutions: there is nothing to give a scale value to the answer. However, the fact that the probabilities must themselves sum to 1 gives a unique solution.

The equations have an important property, which is that the coefficient values are all easily determined from the actual number of an equation. In fact one only needs to hold one number for each equation, the residual error, and apply a relaxation or linear iteration method. This was pioneered by Elldin and Wallstrom at LM Ericcson in Sweden.

So there are three apparently distinct ways of finding the blocking. How do they compare in real situations? When the number of interconnections gets realistic, an Erlang-type formula becomes harder to apply. It was the evaluation of one of these, due to Adelaar, for a range of values which got me going on Zebra — on an all night run at Newport. The equations of state also get very large in number, but simulation remains feasible.

An important point with simulation is that various rules for selecting the actual circuit can easily be accommodated, whereas a formula often applies only to a case where random selection is assumed, leading to an underestimate of the capacity.

The disadvantages of simulation, though, are the lack of insight that is provided by a formula plus the endless computer time needed. When Martin Lawn and I were simulating an early version of the TXE4 series of exchanges, we left runs on the Zebra overnight and sometimes all weekend. Occasional store parity errors did not matter because the program was arranged to restart at the next event; at the end of a run the simulated exchange was cleared of calls and any remaining fragments printed out. I can only remember one such case in fact.

And how about the problem of getting simulation programs correct? We put the whole lot in the store and let it rip, relying on lots of internal checking. For example there were supervisory records held about calls in progress so that the circuits they used could be freed when a call was released; when a new call was made there was a check that the place was empty where a new supervisory record was to go. From time to time we counted the number of calls coming INTO and OUT OF a switch to make sure they agreed.

The program only ran for a fraction of a second or so at first before encountering a failure. During development it once continued for an hour or so before reaching a worrying halt which turned out to be the simulated subscriber calling himself. This had not been thought of and catered for properly.

At one stage, Jacob Kruithof of BTM sent me a set of 64 equations arising in a trunking study hoping that the Zebra could solve them. Although there were about a dozen Zebra library programs for solving simultaneous linear equations, most of them did not handle 64 equations.

Philip Cross and I set about making a program which would use the whole store for working space without calling on the LOT at all. It could solve 114 equations, but its structure became messy with patches.

Hardware Diagnostic Programs

I would like to end by discussing two test programs for Zebra hardware. The first detected open-circuit wiring in the read/write heads on the drum, and the second tried to identify which component was faulty in the parallel multiplier of the transistor machine which followed Zebra.

First the open-circuit test, which I am describing so as to illustrate Zebra's fast track switching. Imagine that you are faced with locating an intermittent open circuit among 256 wires. There was a program which took a minute or so to prove whether or not all the store tracks functioned, but it was too slow to be used while moving the wires around, or tapping the track selector switch.

The easy part was writing a pattern in every 34th word, including the tracks where the test program itself was stored. This wrote at the rate of one track every other word time, or 1600 tracks per second, but overlooked 15 tracks. The checking part managed to read from one track every six word times so the the overall time to test all but the overlooked tracks worked out at 0.6 seconds. When the remaining 15 tracks were included it took 0.7 of a second overall.

It was an unpopular program to run because, despite its effort to save all the locations it was going to write into beforehand, it usually left the store in a corrupt state. Perhaps it always had errors.

The Zebra's successor machine was ordered in early December one year. "Can anyone remember filling in a pink slip for a new computer?", we were asked. No one could, so we filled one in and received a transistor Zebra on 30 December, in time to balance the financial books.

We did not have the extra ferrite core store or the parallel multiplier on our machine, but did develop programs to use them and ran them on an STC machine. I was struck by the difficulty of deducing which multiplier card was in error from the results of its test program, which just printed a multiplier, a multiplicand and a product.

The diagnostic program evolved from an experiment to simulate logic circuits which used one "multiplier brick" as an example. Two bits are resident as multiplier bits; two flow forwards from the multiplicand and two flow backwards from the partial result. Thus six bits are input to the brick. I chose arbitrarily to collect four output bits for each case of the six input bits, ie 256 bits in all which formed a signature of a good brick.

After this it was straightforward to create a list of 68 faults which were applied one at a time, and collect all the output signatures. The run took 11 hours. I worried whether the output was right and repeated it. There was one difference but it was quite clear which was correct since all outputs had to be positive and the bad one was negative.

Elimination of duplicates led to 37 distinct fault patterns. A few faults stimulated both sides of a flip-flop and therefore gave uncertain outputs. Unsqueamishly I ignored these cases altogether and such faults made no entries in the "fault dictionary".

It never crossed my mind to simulate double faults, but then it might have needed 68 different 11-hour runs, and there would not have been enough space to hold the fault dictionary.

The hard part was applying the patterns to the real multiplier since the brick situated two to the left of the one under test had to be set into a “carry” condition occasionally so as to reflect the right pattern backwards. Sometimes two steps were needed to find the right output.

During my last days at STL I went to the factory at Chase Side where transistor Zebras were being assembled and tried the program on six bricks which were faulty. It reported a known error in four of them and an unknown one in the other two — presumably a double fault. The drawings showing the locations of the components on the brick were frustratingly unavailable, but I had little doubt that the reports were correct.

What happened next? There were problems getting the transistor Zebra to run for more than a couple of minutes at a time, and although with hindsight it was obvious that mains interference was the cause of it all, it was not until Vince Fisher came to look at the problem and installed a motor alternator set two and a half years later that it ran properly.

After 10 months or so without being able to run it I wrote a memo to STC. “It has come to my attention that spots of rust are forming on the free running rollers of the Elliott readers on the STL Harlow Stantec System. Normally I should write to Elliotts for advice on how to prevent this rust from forming, but feel that it is correct to address the question to you in the first instance. Anyhow STC has more customer experience of this situation than Elliotts”.

“Can you advise Don?” is marked on the memo. And, lower down, someone has written “Would tape coated with emery dust help?”.

Editor’s note: this is an edited version of the talk given by the author to the Society during the half day seminar on the Zebra computer at the Science Museum on 11 October 1995. Don Hunter was in charge of the Zebra installed at STL in Harlow from 1960-1963, when he left to join CAP (Computer Analysts and Programmers). He is currently a computer contractor.

Experiences with Pegasus 1

Donald Kershaw

The author, a mathematician, was a user of the Ferranti Pegasus with two different organisations from 1956 to 1964. This article describes his memories of those days.

Pegasus was first known as FPC I, with FPC standing for Ferranti Package Computer. Ferranti's usual practice was to avoid the acronyms common at the time — Eniac, Edsac, Univac etc — and use names instead, so it was christened Pegasus. Other Ferranti computers had similar classical names, such as Mercury, Orion, Perseus, Sirius and Atlas.

Pegasus mark 1 had an immediate access store consisting of 48 registers which were built from nickel delay lines. It also had a drum store with initially a 4K word capacity, which was later increased to 7K. The drum on Pegasus mark 2 (and I think on Pegasus 1 also) rotated at 3720 rpm, so one revolution took 16 ms.

Data was read from the drum into the immediate access store in blocks (it was possible to read in single words, but this was very expensive in computer time). There was no need for optimum programming (in contrast to Deuce with its long baths of mercury delay lines), but a slight advantage could be gained by ensuring that the read head was directly above the track you needed to read from.

Input and output was via 5-hole paper tape. Programs were prepared using a Creed perforator. It was usually not possible to tell if a mistake had been made until the tape was run through the interpreter, although certain patterns of holes, such as 11110/01101 for carriage return/line feed, were easily recognised.

Editing was carried out using a hand punch. It involved splicing one piece of tape to another and covering the join with gummed plastic tape. Holes could be punched through the area of the join if required.

Pegasus 1's word length was 39 bits, which was enough to cater for decimal numbers up to 11 digits long. Two orders were stored in one word, with 19 bits to an order. Arithmetic was fixed point: you could perform floating point arithmetic by subroutines, but it was slow. Double length floating point was also available, but it was extremely slow, and was used only for special ill-conditioned programs.

There was a library of subroutines, which was stored on very durable blue paper tape that behaved like spring steel. It contained standard functions such as square root (which occupied just one block) and logarithmic and circular functions, together with some matrix routines. There was also a matrix interpretive scheme, which was a sort of high level language. Later came an autocode which made programming easier, but slowed down the machine. (It was based on the work of Tony Brooker who designed the Atlas autocode, the best programming language I have ever used.)

Here is an example of an autocode program to calculate a sum of squares[1].

```
v2=TAPE (numbers v2, v3,... and set n0=number of numbers)
v0=0
2)v1=v(1+n0) x v(1+n0)
v0=v0+v1
n0=n0−1
→ 2, n0≠ 0
PRINT v0 (print sum of squares)
v0=sqrt v0
print v0 (print square root)
STOP
```

I did almost all of my programming in machine orders with fixed point arithmetic. This was very inefficient in human time but efficient in machine time. It often led to scaling[2] problems, particularly in the solution of differential equations. I still remember the irritation at seeing the OVR (Overflow Register) light showing on the console.

My first experience of Pegasus was gained at Vickers-Armstrong (Aircraft) in Weybridge, which I joined in July 1955. The works occupied the site of the disused Brooklands motor racing track, opposite the fashionable St George's Hill estate. It has now closed down, but there is a museum on the site.

At the time the firm was building the last of the Valiants and the first of the Vanguards. Barnes Wallis had a small research team in what was known as the Chinese Tea House — they were designing the swing-wing aircraft, I believe. There was also a guided weapons group on the site, working on a device codenamed "the Red Dean". There is an example of

[1] Taken from the Pegasus 2 programming manual by G Felton.

[2] Scaling meant that numbers were scaled so that the result of an arithmetic operation was always in the range $-1 \le x < 1$. Division was always a hazard, but multiplication was a safe operation.

one at RAF Cosford.

I was recruited to help form the mathematical services group. The driving force behind the group was Harry (HPY) Hitch. He was very energetic, a keen hockey player, and also had other interests: he was a member of the local D'Oyly-Carte Operatic Society.

Harry was the man who persuaded Vickers, after a visit to the US, to buy a computer. I was told by Donald Davies (not the NPL man, but a former colleague at Vickers) that the management wanted to buy an English Electric Deuce because there were connections between the two companies. It was probably Harry who swung the decision towards Pegasus.

I was at Vickers for only six weeks before my health broke down, and I had to go into hospital for a year. When I returned I found the mathematical services group was in operation under the leadership of Gerry (WG) Moorehead. Donald Davies and John Elliot were senior mathematicians, and there were some key punchers who prepared data on the day-to-day running of the aircraft in various airlines. Gerry and John eventually left to join CERN in Geneva: Gerry remained there, though John subsequently rejoined Vickers. Donald Davies went into teaching when Vickers closed down.

The company had ordered a Pegasus from Ferranti, but had to wait its turn in the queue. I was sent on a training course to learn programming at Ferranti's bureau in Portland Place, where the very first Pegasus was housed. I think George Felton was one of my teachers.

Back at Weybridge we were given problems to program. The Viscount was now in production, and work was also carried out on the Vanguard.

My first task was concerned with refuelling. I remember very little about it except that the central subroutine was to find the positive intersection of two concentric ellipses. But I do remember my pleasure when I finally got my first subroutine right. The solution was printed out to four or five decimal places, and it agreed with my hand calculations.

The subroutine was incorporated into the master program, and after a time (probably a long time) I started my production runs. I was annoyed to find that the engineer who gave me the problem, Albert Pruden, did not believe my results. Since he was in charge I had to re-check the program, and I found that the central subroutine was wrong. Had I checked it to eight decimal places the error would have been obvious. What I had done was to use the order 01 (add into the accumulator) instead of 03 (subtract from it). In the test example the numerical difference was negligible to five decimal places.

This is a lesson I have never forgotten, and the routine of checking to the full accuracy of the machine has been impressed on all I have taught programming. I also learnt to trust Albert's judgment.

The second problem was concerned with the anti-icing system for the Vanguard. This was essentially a tabulation problem. It relied on being able to find the stagnation point of an aerofoil. Once again this was written as a subroutine, and after months of testing it went into production the week before I left Vickers. The program did not work!

Frantic work revealed that when the stagnation point of the flow coincides with the point where the chord line met the leading edge, the subroutine would fail. Fortunately this was easily put right. Anyone familiar with machine language coding will know how non-user-friendly it is, even to the author of the program. If the first production run had not needed this special case the error would have lain dormant for someone else to unravel. It might have been quicker to have rewritten most of the routines.

While Vickers was waiting for its own machine it was necessary to go to Portland Place and use the Pegasus there. I used to clock on in Weybridge at 0830, work there for a few hours preparing tapes on the Creed punch, and then go up to London. Time was scarce, and we booked the computer by the minute.

Since there were many users we could get as little as three minutes on the machine, followed by 10 minutes to find the inevitable error and then a wait of perhaps an hour and a half to get on the machine again. I have never been a careful programmer, and I expected the machine to tell me when I was wrong rather than working meticulously through the program.

Single-shotting through a program was strongly discouraged as being time-wasting. (Single-shotting meant that the orders were obeyed one at a time by operating a switch. The operation of the machine was slowed down so that the registers could be examined in the display oscilloscopes at leisure.)

I met Derek Milledge at Portland Place, and remember his rather weary response to claims that the program was right and it must be the machine that was making the mistakes. In my experience of using Pegasus for over eight years, I know of only one instance where it did make a mistake. The Pegasus order code was very user-friendly and easy to remember.

I never used the main rival of Pegasus, the English Electric Deuce. This was the production version of the NPL Pilot Ace. I was told that there were two Deuces at RAE Farnborough, and important programs were run in tandem on them. If the results agreed they were deemed to be correct! This was unnecessary with Pegasus.

I recall being shown the Deuce in the mathematics division at NPL some years later, and was shocked to find that the usual method of clearing the drum was to open the casing and draw a magnet across it! The magnet was covered by a cloth to prevent it scoring the surface accidentally. But the visual display was superior to that of the Pegasus, and there were some impressive demonstration programs at NPL.

I would clock off when I reached my lodgings in Walton-on-Thames, which was often well after midnight. So my standard of living fell dramatically when Vickers finally got its own machine. This was in early 1957. It was, I believe, the sixth Pegasus to be built, and may well be the one now in the Manchester Museum of Science and Industry.

The machine was used by Vickers Supermarine for their calculations. I remember one of their staff whose programs always worked first time. We consoled ourselves with the thought that he must have taken many hours to check the program thoroughly, but I secretly envied his skill.

When I first joined Vickers I had a desk in the large acreage of the general drawing office, but I returned from hospital in 1956 to find that a new office had been built next to it, with the machine room next door. We could then single-shot as much as we liked.

I was able to do some mathematics, but not enough to keep me interested, and when I told Harry that I had been offered a job in the Civil Service, he did not react in the usual fashion by offering me an increase in wages, but accepted that I was in the wrong job and wished me well.

I do not regret my time at Vickers. I learnt a great deal, and I have always loved aeroplanes. One of the pleasures of working there was to walk round the large construction shed during the lunch hour.

I left Vickers for the Admiralty Research Laboratory (ARL) in 1957, where I joined the Mathematics Group. The atmosphere was completely different: it was almost a research institute.

The Group leader was Stephen Vajda, who is now well into his nineties, but is a professor at the University of Sussex. Until recently he was still teaching and writing books. He was one of the leaders in operational research, and wrote books on the theory of games and on linear programming.

Fortunately for me, ARL also had a Ferranti Pegasus. Dr EM Wilson introduced me to numerical analysis, and I solved my first differential equation there, but on computing paper. The work at ARL was more mathematical than at Vickers, and we were encouraged to carry out personal research if it seemed to be connected to Naval interests.

Probably the most important series of programs that I wrote were concerned with the design of the nose profile of an acoustic torpedo. These had flat noses, which lead to problems of detachment and thence to turbulence. IJ Campbell of Underwater Research Establishment was in charge of the project.

It involved the solution of a Fredholm integral equation of the second kind for the speed of the flow of the fluid round the profile. The equation had been derived by F Vandrey, a German fluid dynamicist.

Before my arrival the problem had been attacked by a team of computors (a term used in those days to describe scientific assistants) who used hand calculators. These were almost certainly Brunsvigas taken from the German navy at the end of the war: there were Marchants at ARL, but I do not remember them being used extensively.

I was told that there was a steady flow of people into and out of the group that worked on this problem. I have every sympathy with them. Vandrey did his best to simplify the problem, but there was an enormous number of bread and butter calculations. Each had to be done twice to avoid the mistakes that were an inevitable consequence of the boring and repetitive nature of the work.

The *ad hoc* method for solving the torpedo problem was to find the flow round a series of profiles and then choose the one that had the optimum speed of flow. When the maximum speed was lowest, then according to Bernoulli's equation the pressure was highest. This reduced the likelihood of a flow being detached. The best profile was tested in a wind tunnel, and led to full scale torpedos being built.

By that time the store of Pegasus had been increased from 4K to 7K 39-bit words, but I needed much more than that. I had to divide the total job into five distinct programs. When one had finished the next was read in. The final program had to solve a set of about 70 linear algebraic equations.

All arithmetic was fixed point, and the programming was done in machine code. I think it took me about a year to complete the programs.

Production runs were often made on Saturday mornings. I would switch on the machine, hope that the reading heads did not grind into the drum as it came up to speed, set the first program in action and read a book until it printed out the results three hours or more later.

In a sabbatical year recently I decided to re-program the problem, but this time using an optimisation program. Each stage consisted in setting up and solving the integral equation followed by a calculation of the maximum velocity which was minimised. The programming only took a few weeks, and the central routine was now the whole ARL program which would be gone through many times in one complete run.

The results were interesting. They showed that the optimum profile for the nose had a bulge like the nose of a leek. This made it impossible for it to be fired from a submarine. It was the best profile in theory, but useless in practice. I told both the Admiralty and the US Navy about the results: they were polite but indifferent.

The other problem I remember clearly was that of determining the sound field propagated by the propellor of an atomic submarine (the Valiant, I believe). This was done in collaboration with DE Ryall and Kendrick of the hydrodynamics section. I did not take this very seriously, and only wrote a few preliminary programs.

Then one day in early 1962 I was told that someone was going to America shortly and wanted to show the results to Admiral James. There was no time to automate the rest of the calculations, and I had to carry them out by hand on one of the captured German Brunsvigas. Fortunately I completed them in time.

The last time I was at ARL was just before the buildings were taken over by the National Physical Laboratory. Jimmy James (a former colleague and the last surviving member of the Mathematics Group) and I searched unsuccessfully for a souvenir Brunsviga. They were the best machines of their type.

ARL is now closed down. I enjoyed working there, and learnt what mathematical research meant. It is not possible these days to have such a relaxed atmosphere. In fact it disappeared shortly after my time there. Dr Vajda left the year after I did to take the chair of operational research at Birmingham University, and the emphasis thereafter switched to statistics.

Other members of the Mathematics Group were Tom (TG) Weale, Beryl Kitz (who had worked at GCHQ) and Martin (EM) Beale, who became an FRS: he died a few years ago.

Jack Good, formerly a member of the Bletchley Park Ultra scheme, was also a colleague for a few years. He transferred from GCHQ. He is now in Virginia and has, I expect, retired from teaching. I believe that he had been a research student of GH Hardy at Cambridge. He is a prolific mathematician: his main area is probability theory.

I left ARL in March 1964 to help form the Computer Unit at Edinburgh University, to teach numerical analysis. I never became interested in computers or computer science. Sometimes I regret not having got in on the ground floor, since under the leadership of the late Sidney Michaelson Edinburgh has a high-ranking Department of Computer Science. But I have always treated computers as a means to an end of solving problems and not as an end in themselves.

Editor's note: This article is an edited version of the talk given by the author to the North West Group of the Society at the Manchester Museum of Science and Industry on 12 April 1994. Donald Kershaw is Reader Emeritus in the Department of Mathematics and Statistics at Lancaster University.

Internet addresses

Readers of *Resurrection* who wish to contact committee members via electronic mail may do so using the following Internet addresses.
Chris Burton: chris@envex.demon.co.uk
Martin Campbell-Kelly: mck@dcs.warwick.ac.uk
Dan Hayton: Daniel@newcomen.demon.co.uk
Adrian Johnstone: adrian@dcs.rhbnc.ac.uk
Tony Sale: t.sale@qufaro.demon.co.uk
Robin Shirley: r.shirley@surrey.ac.uk
Doron Swade: d.swade@ic.ac.uk

Letters to the Editor

Dear Editor,

I was amazed to read CP Marks' letter in the autumn issue, which referred to a demonstration of Cafs (content addressable file store) from ICL (then ICT) as long ago as 1968.

I worked in public relations for ICL in 1978, when this trail-blazing product and the Distributed Array Processor (DAP) were tentatively being demonstrated. It seemed that ICL just could not decide what to do with these two world leader products. Yet CP Marks' letter suggests that the company had already been dithering over Cafs for 10 years.

Cafs is now a standard product but sadly the DAP never really caught on while ICL was handling it. Perhaps this was down to the company's short-sightedness: perhaps the world was not ready: in 1980 I asked a senior man close to the DAP why ICL did not shout about this marvellous machine, and he replied, "The trouble is, it doesn't run payroll".

Yours sincerely,

John Kavanagh
Croydon
12 September 1995

Dear Mr Enticknap,

Thank you for publishing Gordon Scarrott's accounts of ICL research projects, which I read with great interest.

One invention which he didn't mention, and for which I think he deserves credit, is that of slave stores. He introduced me to the idea early in 1963, and we discussed it at some length. I was able to visit Rice University in 1964 to make what I think were the first practical measurements of program behaviour with slave memories. It was not obvious at the time how to present the results: the aim was to measure performance gain, but we ended up displaying the 'hit rate'. The performance of the Titan slave store, which went into service shortly afterwards, provided a sharp reminder of the distinction.

I think the first published account was by Wilkes in 1965, and the first successful implementation was in the cache memory of the IBM System/360 Model 85 in 1968. At the time things seemed to happen very slowly, but in the light of subsequent events this was a remarkably rapid transfer from laboratory to commercial product.

Yours sincerely,

John Iliffe
London N2
18 October 1995

Dear Mr Enticknap,

In issue 10 I relayed information from former staff at Vickers Armstrong Aircraft that it was Pegasus number 6 which was donated by Vickers to Brooklands Technical College around 1965, and later acquired by the Manchester Museum of Science and Technology.

In his article in issue 12, Ken Turner concluded that the packages in the Manchester machine originated from Pegasus number 1, but that the machine itself was number 6. At first this seems strange, but there was a good reason for such an exchange of packages.

In 1965 Vickers Armstrong Aircraft at Weybridge had three Pegasus computers, with serial numbers 1, 6 and 33. As work was transferred to the company's new ICT 1905, it was possible to release one Pegasus.

When Pegasus number 1 had been at Ferranti's first London Computer Centre in Portland Place, it was used to test the large drum and the character handling instructions that were later incorporated in Pegasus mark 2. As a result of these enhancements, Pegasus number 1 was fully compatible with the Pegasus number 33, a mark 2 machine, though with the Intermediate Access Store the latter was faster. Pegasus number 6 did not have the enhancements and so could not run all Vickers' programs, so it was chosen to go to Brooklands Technical College.

The packages originally in Pegasus number 1 were prototypes, some two years older than the production packages in number 6. To the maintenance engineer, responsible for a reliable service at Vickers, it would have seemed best to keep the production packages and transfer the prototypes to number 6 before he moved it to Brooklands Technical College.

Yours sincerely,

Derek Milledge
Bracknell
Berkshire
27 November 1995

Forthcoming Events

13-14 January 1996, and fortnightly thereafter Guided tours and exhibition at Bletchley Park, price £3.00, or £2.00 for concessions

Exhibition of wartime code-breaking equipment and procedures, plus 90 minute tours of the wartime buildings.

6 February 1996 North West Group meeting

The Origin and Development of Database Software, by Professor Peter King.

February 1996 London meeting

The Atlas computer: more details when we have them.

23 April 1996 North West Group meeting

Industrial Research in the Information Technology Field, by Gordon Scarrott.

21 May 1996 North West Group meeting

The Small-Scale Experimental Machine Rebuilt, by Chris Burton

May 1996 Whole day seminar

The ICT/ICL 1900 series: more details when we have them.

The North West Group meetings will be held in the Conference Room at the Museum of Science and Industry, Manchester, at 1730 hours. Refreshments are available from 1700.

Queries about London meetings should be addressed to Chris Hipwell on 0182572 2567 or George Davis on 0181 681 7784, and about Manchester meetings to William Gunn on 01663 764997.

Resurrection is the bulletin of the Computer Conservation Society and is distributed free to members. Additional copies are £3.00 each, or £10.00 for a subscription covering four issues.

Editor – Nicholas Enticknap
Typesetting – Nicholas Enticknap
Typesetting design – Adrian Johnstone
Cover design – Tony Sale
Transcripts of talks – Pat Woodroffe and Hilma Quinn
Printed by the British Computer Society

Committee of the Society

Chairman **Graham Morris FBCS**, 43 Pewley Hill, Guildford, Surrey GU1 3SW. Tel: 01483 566933

Secretary **Tony Sale FBCS**, 15 Northampton Road, Bromham, Beds MK43 8QB. Tel: 01234 822788.

Treasurer **Dan Hayton**, 31 The High Street, Farnborough Village, Orpington, Kent BR6 7BQ. Tel: 01689 852186

Science Museum representative **Doron Swade CEng MBCS**, Curator of Computing, The Science Museum, Exhibition Road, London SW7 2DD. Tel: 0171-938 8106

Chairman, Elliott 803 Working Party **John Sinclair**, 9 Plummers Lane, Haynes, Bedford MK45 3PL. Tel: 01234 381 403

Chairman, Elliott 401 and Pegasus Working Parties **Chris Burton CEng, FIEE, FBCS**, Wern Ddu Fach, Llansilin, Oswestry, Shropshire SY10 9BN. Tel: 01691 791274

Chairman, S100 bus Working Party **Robin Shirley**, 41 Guildford Park Avenue, Guildford, Surrey GU2 5NL. Tel: 01483 65220

Chairman, North West Group **Professor Frank Sumner FBCS**, Department of Computer Science, University of Manchester, M13 9PL. Tel: 0161 275 6196.

Chairman, Meetings Sub-committee **Christopher Hipwell**, Bretts, The Green, Newick, East Sussex BN8 4LA. Tel: 0182572 2567

Secretary, Meetings Sub-committee **George Davis CEng FBCS**, 4 Digby Place, Croydon CR0 5QR. Tel: 0181-681 7784

Editor, Resurrection **Nicholas Enticknap**, 4 Thornton Court, Grand Drive, Raynes Park SW20 9HJ. Tel: 0181-540 5952 Fax: 0181-715 0484

Archivist **Harold Gearing FBCS**, 14 Craft Way, Steeple Morden, Royston, Herts SG8 0PF. Tel: 01763 852567

Dr Martin Campbell-Kelly, Department of Computer Science, University of Warwick, Coventry CV4 7AL. Tel: 01203 523196

Professor Sandy Douglas CBE FBCS, 9 Woodside Avenue, Walton-on-Thames, Surrey KT12 5LQ. Tel: 01932 224923

Dr Roger Johnson FBCS, 9 Stanhope Way, Riverhead, Sevenoaks, Kent TN13 2DZ. Tel: 0171-631 6388

Dr Adrian Johnstone CEng, MIEE, MBCS, Department of Computer Science, Royal Holloway and Bedford New College, Egham, Surrey TW20 0EX. Tel: 01784 443425

Ewart Willey FBCS, 4 Sebastian Avenue, Shenfield, Brentwood, Essex CM15 8PN. Tel: 01277 210127