

# 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
- ◇ Develop awareness of the importance of historic computers
- ◇ Encourage research on historic computers

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.

# Resurrection

The Bulletin of the Computer Conservation Society

ISSN 0958 - 7403

Volume 1 Number 4

Summer 1992

## Contents

### Editorial

*Nicholas Enticknap, Editor* 2

### Guest Opinion

*Director, The National Museum of Science and Industry* 2

### Society News

*Tony Sale, Secretary* 4

### The early days of Algol

*Nicholas Enticknap and Pat Woodroffe* 5

### The influence of Alan Blumlein on early computers

*Ted Newman* 10

### The pre-history of the Digital Equipment Corporation

*Adrian Johnstone* 13

Early computers at Manchester University 20

Miscellany 26

Working Party Reports 27

Forthcoming events 32

---

## Editorial

*Nicholas Enticknap, Editor*

---

It has been a long time since the last issue, and much has happened. The most significant development has been Tony Sale's initiative in setting in motion plans for a Museum of Cryptology and Computing in Bletchley Park, on the very site where the Colossus code-breaking computers played such a vital role in World War Two. Tony summarises the current situation in his Society News piece.

We hope this enterprise will eventually result in both a permanent home for the Society and its machines and other artefacts as well as a focal point for our various activities. These have expanded with the arrival of our Archivist, Harold Gearing, who has started work on the massive undertaking of documenting and classifying our growing collection of historical documents.

Meanwhile the Society's existing activities have continued unabated. As far as work on our two oldest computers is concerned, the emphasis has changed — both Pegasus and Elliott have been working for over a year now, and the new objective is to get them back to the state of reliability they enjoyed when new.

The two working parties concerned with more modern computers have both been expanding their portfolios of equipment, while the Software group has taken on a new lease of life with a redefinition of its role. Further details of all this activity can be found in the Working Party reports.

The meetings programme has continued throughout the past year along the lines successfully established in our first year. This issue carries reports on a wide variety of historical subjects, ranging from the evolution of Algol to the influence on early circuitry of Alan Blumlein via the development of the DEC PDP-8. We also carry a report based on the all-day seminar on the pioneering computers from Manchester University.

---

## Guest Opinion

*Director, The National Museum of Science and Industry*

---

The activities of the Computer Conservation Society have agitated a host of issues that are both exciting and disturbing. Several established tra-

ditions have been challenged. There is an unspoken assumption in the museum world that interpretation of the past is necessarily based on partial, if not fragmentary, evidence. However, the Society's programme, far from being damned to shards and chipped fragments from an inaccessibly distant past, provides an embarrassment of riches — machines, documentation, orally transmitted expertise and experiences, recorded seminars and colloquia by the living practitioners and witnesses to the great pioneering age of computing. Unlike the archaeologist we have the opportunity to select from an almost unlimited wealth of evidence.

There is another respect in which the activities of the Society up-end otherwise comfortable perceptions. The established ethos of preservation is essentially passive — physical relics placed in an inert environment to retard degradation. However, the efforts to restore historic machines to working order by those who originally designed and maintained them are highly interventionist and this raises dilemmas about historical integrity and authenticity.

These are not the only stirrings in the museological soup. It is painful to accept the ultimately inevitable demise of machines lovingly and ingeniously restored to working order. The Society's programme to emulate past performance on present day machines using the restored original as a benchmark of authenticity, is a bold and visionary move, but it has disturbing implications. The museum culture is founded on the notion that the original object is an ultimate historical source. However, with emulation, we are faced with the prospect of the operational spirit of a machine being preserved, as it were, independently of its body. The 'ultimate source' is now a computer program not an object. Can museums extend to embrace the abstract as object?

These are a few of the many rewarding speculations stimulated by the activities of the Society which has broken new ground in conservation and museological terms. Its accomplishments are to be saluted and we are indebted to its many members for the generosity of their efforts and time, The Science Museum cherishes its relationship with this vigorous and unconventional organisation and looks forward to building on its successes and progressing its pioneering programme in the coming years.

---

## Society News

*Tony Sale, Secretary*

---

We spent a lot of time last year working on the long term future of the Society. We have been concerned that our future is anything but secure. Our premises at the Science Museum are already too small for our growing collection of computers, and in any event the building is under threat of re-development.

Last summer a unique opportunity arose to secure a permanent home for the Society. This is in the most appropriate site of Bletchley Park, where the Colossus code-breaking computers were developed and used during World War II.

Bletchley Park is currently still largely owned by the Government, represented by Property Holdings, with the remaining small part belonging to British Telecom. The two organisations were planning to sell the Park for re-development, including the huts used by the code-breakers and a number of other buildings used to house teleprinters and other equipment.

The Society joined forces with the Bletchley Archaeological Society to lobby the appropriate local authority, the Milton Keynes Development Corporation, for support in preserving the historic huts. On our own initiative we further proposed that the part of the Park which includes the huts should be used to house a Museum of Cryptology and Computing. The proposed Museum site covers four acres, and contains 9,000 square feet of space in the huts and a further 60,000 square feet in the other buildings, which are currently leased by the Civil Aviation Authority.

I am glad to say that our efforts have been successful. The Milton Keynes Development Corporation has set up a Bletchley Park Trust with a seedcorn start-up fund of £20,000. The Trust is currently raising further funds from industry.

The Trust has a steering committee, and has produced an outline business plan. This involves turning the whole of Bletchley Park into a 1940s theme park, of which the museum campus will form a part. Income from the theme park will provide the funding for the Museum.

There is a long way to go yet. The Trust needs to raise £10 million to purchase the Park. Nonetheless, I am very hopeful that the project will succeed, and remove the uncertainty hanging over the Society's future.

---

## The early days of Algol

*Nicholas Enticknap and Pat Woodroffe*

---

The first public implementation of Algol was created in January 1960. The thirtieth anniversary of the language, therefore, took place in 1990. To commemorate the occasion, the Society held an afternoon meeting at the Science Museum.

The meeting concentrated on the early days—how Algol came into being, and how the early compilers were developed. It was chaired by Mike Woodger, a member of committees responsible for defining the language as far back as 1958. Woodger gave a scene-setting talk describing the creation of the language.

Broadly speaking, Algol emerged from the activities of two independent groups: the GAMM subcommittee in Europe, and the ACM in America. The major steps in the process are listed in the following table:

Date	Event
1955 Oct	GAMM subcommittee for Programming Languages formed
1957 Oct	ACM committee appointed to specify a universal programming language
1957 Oct	GAMM Subcommittee's language design finished: letter sent to ACM suggesting a joint conference
1958 Apr	GAMM proposal presented to ACM group
1958 May	Joint ACM/GAMM proposals made (the language was later called IAL or Algol 58)
1958 Oct	The name "Algol" is proposed for the new language
1958 Nov	First Algol 58 compilers reported working
1960 Jan	Algol 60 created in Paris
1960	Dijkstra and Zonnefeld produce first Algol 60 translator.

Woodger also ran over the genesis of the various ideas and concepts that found expression in the early implementations of Algol, starting as far back as 1948.

Woodger was followed to the rostrum by David Hill of the Medical Research Council, who took the story forward into the seventies with an account of subsequent standardising work by IFIP and ISO.

ISO and IFIP both entered the picture in 1962, when IFIP Working Group 2.1 took over running Algol as ISO was turning its attention to

the standardisation of programming languages. So the Working Group submitted its 1962 Algol report to ISO.

ISO, said Hill, decided the report wasn't suitable as it stood. It wanted a subset, some I/O to go with it, and a standardised hardware representation of the language.

In response the Working Group produced by 1965 three subsets of the language plus two proposals for I/O. These were put together in a Draft Recommendation: ISO decided not to call them standards because there were still arguments about whether there should be one standard language rather than standards for various languages.

The draft appeared in April 1967, but it was March 1972 before scrutiny of the draft was completed and 'ISO recommendation No 1538: Programming language — Algol' appeared.

This, said Hill, "was an absolute disaster, full of errors and changes which had been made without consultation". Hill himself was instrumental in rectifying the "disaster", along with Richard de Morgan and Brian Wichman. Together these three formed a new IFIP sub-committee to consider further revisions to Algol 60.

This sub-committee had a number of sources to draw on. At a 1972 ISO committee meeting it was agreed that a new edition of ISO Recommendation 1538 should be produced. The delegates at this meeting drew up a list of errors in the earlier ISO document, and also drew up a list of questions that ought to go to the new IFIP sub-committee. There were also proposals from German and Japanese sources, and a number of published papers by Algol specialists.

The IFIP sub-committee was finally left with a number of difficulties of interpretation, plus some unsettled questions. On these points the committee had to make its own decisions.

Once all the difficulties had been resolved, the maintenance committee's report was agreed by IFIP Working Group 2.1 and subsequently became known as 'The Modified Report'.

At this point the emphasis of the Society's meeting shifted from the specification to the implementation. The next three speakers described the development of compilers for three sixties computers, the English Electric KDF9, the Elliott 803 and the DECsystem-10.

Lawford Russell joined English Electric in 1960, and after initial work on the Deuce teamed up with Brian Randall to produce an assembler for the KDF9, then under development. Before the project made much

progress, however, the two men went to an Algol conference and became “very excited by it all”.

Deciding on the spot that they must implement Algol on the KDF9, they approached Algol guru Edsger Dijkstra for advice. The upshot was that they spent an intensive two weeks study with Dijkstra in Holland, which “led us to an outline of the basic structure of the intermediate language, which we transposed Algol into for running”.

Randall and Russell made two early important design decisions. The language would be interpretive, as that was what the users in their division of English Electric wanted. And “we decided to go for a single pass translation as we were used to on the Deuce, which allowed you to do some processing while your cards were being read in”. The KDF9 would be faster and would use paper tape, so “surely we could do some processing for the translation while the program was being read in?”.

The striking aspect of this project was the working conditions which would be thought impossibly primitive today. “We had no machine and very little information about it so we had to do lots and lots of desk checking. We produced our logic flow diagrams and just desk checked these between us... So the logic checking got far ahead of the code testing: some of that was good and some of it was bad.”

About this time potential KDF9 customers starting asking for copies of the compiler. But it wasn’t finished, and “our flow diagrams were only meant for inter-project use. So we thought we would need to define a much more rigid sort of meta-language... then we had to convert all our flow diagrams to this, then we had to desk check them again”.

The logic flow diagrams were actually delivered to one potential customer, de Havilland — “the first case of a copy of a compiler being available before the original”.

The design of the Algol compiler for the Elliott 803 presented a different set of problems, described by Jeff Hillmore. Late in 1960 a small group from Elliott went to an Algol Conference run by Dijkstra — that man again. They returned determined to make Algol 60 the advanced programming language for the 803. A development team was formed, and Hillmore joined it in November 1961.

The team worked with some despatch. The first program was compiled and run just three months later, on 15 February 1962. Version 1 of the compiler, using five-hole paper tape input/output, was available by November. The time taken to reach that point, which included writing



the user manual, was eight person-years.

“The 803 sold to a lot of educational establishments and to scientific users, and in both of these areas there was a need to process a high average number of programs in a given period, and to reduce to a minimum the time spent in changing over from one program to another.

“So the design decisions that were made were, first of all, the system should be a load-and-go system. You should, at the start of a session, load the compiler, load the dynamic routines and the process, compile and run program after program.

“The other decision that was made was that the user’s program should run at the full speed of the machine. So we wanted a compiler that would generate an object program in binary machine code which would then run, and we didn’t want to interpret them. This was a very different decision to that made by the KDF9 people.

“The other decision was that the source text should be read once only. So we ended up with a compiler which was designed to operate in two phases. The compiler converted the Algol source to an intermediate binary stream which we called Own Code, and this was processed in the second internal pass into a binary object program, and that was then executed at full speed.”

Hillmore identified six activities in the compiler writing process. First there was the writing of a lexicon analyser to tokenise the Algol symbols, taking the hardware representation and generating these tokens, which made the compiler independent of how the program was input.

Next came the dictionary system for recording identifiers and their properties. Then the team decided how the memory space was going to be allocated. Fourth, they specified the own code format. “Out of those four parts of the initial activity came the definition of the Algol 60 subset which would be implemented”.

The final two activities were the specification of the I/O definition and the writing of a high level pseudo-Algol definition of the compiler.

The final speaker was Richard de Morgan, who gave a presentation describing the design of the Algol compiler for the DECsystem-10. We have already met de Morgan as one of the IFIP sub-committee that produced the Modified Report. Earlier, he had been one of the earliest users of Algol—he learnt it as his first programming language while an undergraduate at Liverpool University.

de Morgan’s first task at DEC, which he joined in 1969, was to work

on the outline design of an Algol-60 compiler for the DECsystem-10. The only languages supplied for it then were Fortran and BASIC.

de Morgan was working with Professor Nicol Hardiman of Carnegie-Mellon University, and they were given one month to complete the task.

“We did design a system, and the amazing thing is that we could do the design down to a fair amount of detail in one month. It turned out that Nicol Hardiman’s ideas about how the system ought to be built and my experience of writing compilers were more or less compatible.

“We decided that we would do a one pass compiler but we would implement everything we could in what we conceived to be the spirit of the Algol 60 revised report... Of the things we couldn’t implement, if we thought they were useful we’d find a way round them and if we didn’t think they were useful we wouldn’t implement them.”

The decision to build a one-pass compiler created problems. One of the major ones concerned formal parameters, particularly formal parameters that are procedures. The question was: how do you know when you’re going to call a formal procedure parameter what its parameters are?

de Morgan went on: “Given that we had to have independent compilation, the unit of compilation was either the main program or a procedure. So we had the concept of an external procedure: we had to enforce type checking at run-time of the actual to formal parameter correspondence.”

Another design decision was to allow users to do anything they liked with formal parameters. They could even be used as the controlled variable in a forced statement, so that you could predict what was going to happen.

Then questions arose about own variables and own arrays, which were only resolved after much thought.

“We put in a few other extras: position, remainder, operator, which we thought very useful. We put in ‘While’ as an iterative statement. We added another type—non-real—rather than having some type of compiler directive which told you what precision your reals ought to be.”

The resulting compiler worked well. “Efficiency of the code was fairly good because we put work into getting around the overhead of the procedure calls by generating access sequences that were very efficient. We often ran faster than Fortran, and typically 1.5 to three times faster than Algol 60 on comparable machines.”

---

## The influence of Alan Blumlein on early computers

*Ted Newman*

---

I believe that Blumlein was the best electronic engineer that ever lived.

I first met him during the war when, as a young engineer, I saw an advert for someone to do TV tubes and applied for the job. I went along and there was this very frightening man with green eyes who interviewed me—Blumlein seemed to have the greenest eyes of anyone I'd seen.

He started the interview by asking if I knew anything about vector analysis. I said I didn't. He said, "Here are some circuits I want you to look at". He spent a long time making me look at these circuits and asking, "How do you think this works?", "How do you think that works?". Then he said, "Actually I would call that vector analysis. You might think that I'm now taking you to the vacancy we've got in the TV tube place. You're not going to do that. You're going to work on circuits".

I was 22 at the time I started. I was Blumlein's personal dogsbody, and had lots of odd things to do for him. I saw Blumlein much more than I would have expected to, because I was a lad of not much importance. He was always Mr Blumlein to me—never Alan—because he was the big boss.

He created a range of monitors—they were very advanced measuring oscilloscopes. You could measure the amplitude and width of the pulses very accurately. You could do this by shifting the offset voltage on the screen, and measuring the shift on the meter, and the width could be similarly measured with a very accurately defined sawtooth scan.

The first monitor was called Mohammed, after the founder of the Moslems. The Caliphs came after Mohammed in history so the later monitors were called Caliphs—I can remember Ali and Xerxes, but not the others. The monitors were all slightly different and Blumlein expected every one to be correctly named; if you got a name wrong you were in dead trouble.

He also had names for many of his circuits. There was the Cholmondeley Tweaker, a Featherstonehaugh Follower and a St John Something Else. (These names were pronounced Chumley, Fanshaw and Sin-jin.)

He believed that circuits had to be calculated. I had a little book that I'd calculate various circuits in and then he'd see on the monitor whether the pulse was the same as I'd calculated.

Blumlein was born in 1903. He was in Standard International from

1924 to 1929, when he joined Schoenberg's Columbia record organisation, which was a small company at the time. The following year he joined EMI, where he worked till he was killed in a plane crash in 1942.

He filed 128 patents, but he didn't write many papers. If you've got a lot of valuable ideas in a commercial firm you don't want to give them to everybody else, so you don't write papers, you file patents.

Some time after Blumlein's death, in 1947, at an important meeting of the Association of Scientific Workers there was an almost unanimous decision that all research ought to be done in Government and not commercially. I was an idealist at the time so I tried to get into Government service at a very reduced salary.

I went to the Civil Service Commission, and was interviewed by HA Thomas, who asked me "EMI—isn't that where Blumlein was?". I said yes, and he then sent me to NPL.

There was a background to this which I discovered later. At the time Turing had got a lot of details, in fact a full logical diagram, for a both a big computer and a small machine. The big one Turing thought was too much to tackle: the small one was called the Ace Pilot Model. A number of people had tried and failed to get the circuits working at all, including Turing to start with—he wasn't very good at getting circuits going.

I think what happened was that HA Thomas, a much maligned man, thought it would be a good idea to recruit someone who knew something about circuits of the kind that were necessary for computers, and these were the sort of circuits that Blumlein created.

Once I was at the NPL, Thomas asked me to recruit other people with this knowledge. So Tubsy Clayden was recruited, and then later John Parks and Roger Scantlebury (Roger's dad had also worked at EMI for a long time).

The circuits we produced all worked extremely well, and they were derived from Blumlein circuits, so there is no doubting the influence of Blumlein on the very reliable Pilot Model Ace.

Also, Blumlein visited TRE in the forties and met Williams and his colleagues. They had not been doing very well with their circuits, and Blumlein showed them how to do it. Tom Kilburn agrees that this happened. Williams afterwards became Professor Williams who dealt with the Manchester computer, and Kilburn joined him from TRE, and from what I could tell the Blumlein-type circuits were widely used in TRE. So once again the influence of Blumlein as very great.

Blumlein was unique. He was highly inventive, and a man who thought in a different way from his contemporaries. Light engineering in 1929 dealt almost entirely with communication by voice or code. Transmission was by wire cable or radio. I've read all the standard books on communication up to 1945 that I can find, and none of them deals in any way with high speed pulses. Nor, except for the purely transmission side, could they deal with the tasks posed by television.

In the communication industry all control and switching functions were either done mechanically or by electrically controlled relays. High speed was not wanted. It was only recently, long after the advent of computers, that electronics was used. For television and particularly for advanced radars or computers, control and switching had to be done electrically. But Blumlein was an iconoclast and was very inventive: only such a person could break the necessary new ground.

*This article is an edited transcript of a talk given by Ted Newman to the Society at the Science Museum on 28 February 1991.*

---

# The pre-history of the Digital Equipment Corporation

---

*Adrian Johnstone*

---

The Digital Equipment Corporation (DEC) was formed in 1957 and is now perhaps the second or third largest computer company in the world. Remarkably, the company is still led by the original president, Ken Olsen, who was a student at MIT at a key time in computing history. DEC is widely credited with producing the first desktop minicomputer, the PDP-8. The minicomputer style of computer architecture is very different from the long-word length machines that dominated early commercial computing, but the ideas did not spring fully formed from the minds of DEC's designers. I hope to show here that one of the earliest programmable computers constructed in the United States was the direct parent of the DEC PDP-5, -8 and -11 series of machines as well as many early eight- and sixteen-bit microprocessors.

Although the distinctions are nowadays becoming blurred, traditionally computing has been divided into three application areas: *scientific* computing with an emphasis on floating point arithmetic, *commercial* computing which requires access to large databases and *real-time* computing which requires relatively simple processing to be performed at the highest speed. A real-time system must return results within the timeframe of some monitored process — car engine management systems, flight simulators and the computer inside your washing machine are all examples of real-time problems. There is an old joke that defines a scientific programmer as one who types in a single number, processes it for a week and then prints a single number. By the same token, a commercial programmer types in a number and prints out the names of five thousand people that have the same number, and a real-time programmer doesn't type in any numbers because embedded systems do not have keyboards.

In many ways real-time computing is the Cinderella of the three being mainly the province of engineers capable of squeezing the last ounce of performance from computers that have to be low cost, because they are to be embedded in some other product. In the past this meant a minicomputer and nowadays a microprocessor, but in either case the traditions of machine level programming that naturally dominated the pioneering days of computing live on in the real-time domain and there is therefore a strong

affinity between the activities of the Society and the real-time computing world.

Since embedded systems must be cheap, it might be thought that real-time control systems are a relatively recent phenomenon. In general this is true. To be more specific, it was the arrival of minicomputers, especially the PDP-8 in April 1965, that allowed engineers to seriously consider dedicating an entire computer to a single experiment or instrument. However, it turns out that one of the first computers constructed in the United States was a real-time system, and the architectural design decisions that were taken then formed the template for the minis and micros to come. This machine was the Whirlwind.

In 1944 the US Navy contracted the Massachusetts Institute of Technology (MIT) to build an aeroplane simulator that would be capable of solving the equations of motion of the aeroplane in real time and which could thus be used to investigate instability problems in aircraft designs of the day. Originally an analogue machine was proposed, but in 1945 the design turned into a general purpose digital computer. The laboratory at MIT was renamed the Digital Computer Laboratory and Maurice Wilkes who had many contacts with the MIT team, says in his memoirs:

“by the time I knew them they were pillars of digital orthodoxy and I did not suspect they had an analogue past”

The machine was a sixteen-bit single address processor — an arrangement familiar to anybody who has worked with early minis or micros, but in those days to have a machine with only a 16-bit word was heretical. Typical word lengths were in the range 36 to 40 bits because of the need for real number arithmetic precision. No less a figure than von Neumann criticised the Whirlwind saying that with its short word length he was concerned about its ability to do anything useful at all. However, sixteen bits of precision is more than adequate for most sensing and control applications, and if you must have more precision you can always perform multiword arithmetic. Interestingly, on modern machines the word size is dominated by the need to address large amount of memory rather than the size of the data being manipulated: hence the widely expected move towards 64-bit architectures such as the new MIPS and DEC Alpha devices is being driven by the need to directly address more than 4G byte memory spaces, not the need to manipulate 64-bit numbers. I cannot resist pointing out at this stage that von Neumann also had a fight with Edward Teller concerning the UNIVAC LARC (Livermore Automatic

Research Computer) contending that putting anything more than 10,000 words in a computer would *always* be a waste of money.

The real reason that Whirlwind had such a small word size is that it was bit-parallel at a time when almost all computers were bit serial. The Elliot 803 in the museum's collection, for instance, has at its heart a single full adder, through which the two operands are passed one bit per cycle. The Whirlwind processed two sixteen-bit operands in a single cycle, but naturally required sixteen times as much hardware to do so. Since the Whirlwind was being designed for a very specific application, it was possible for the designers to make a detailed analysis of the design constraints. Robert Everett, who was responsible for designing part of the order code for the machine had these comments about how they arrived at a 16-bit, one address computer:

“[Sixteen]... is a nice binary number, but it did not come about arbitrarily. It was determined by asking “What is the shortest single-address instruction that looks reasonable?” Our analysis of the programs we were interested in showed that 1000 words was tight and 2000 considerably better. That gave us eleven bits, and we knew that we needed at least 16 instructions; 32 sounded reasonable, and that gave us five bits more. Therefore the sixteen was not a binary number, it was the sum of two primes.”

Now although I am claiming that Whirlwind was *architecturally* the first minicomputer there is no doubt that the machine was anything but miniature physically. Indeed the scale of the project was staggering. The budget was \$1 million *per annum* for the years 1945 – 1950, which must have given Maurice Wilkes pause for thought on his visits there. The machine was laid out in 2-dimensional form so that every part could be immediately accessed in case of failure. This led to a very space-inefficient design requiring one floor of a large building. The control room alone contained fifteen 6-foot racks along with oscilloscopes and I/O equipment. The machine used specially made valves that cost between \$5 and \$10 each, and the laboratory had its own tube shop to make them. Over 5,000 valves and 11,000 germanium diodes went into the running system. Considering EDSAC had 3,000 valves in it and was really rather compact, it is quite difficult to see why Whirlwind needed so much floor space. Later on large tubes were added to Whirlwind that were used purely for *display* and Whirlwind is therefore probably the first computer to have purpose designed VDU's attached. Whirlwind has an even stronger claim to be



the first computer with a light pen attached.

One important aspect of the Whirlwind design was the instruction decoder. At the heart of the machine was a diode matrix that performed the decoding, rather than a set of random logic gates. This structured decoder was of course only one step away from a microcoded architecture as described by Maurice Wilkes. When Wilkes visited MIT in the 1950's he was already thinking about the use of structured as opposed to random logic and was shown the Whirlwind which:

“... did indeed have a centralised control based on the use of a matrix of diodes... It was not, I think, until I got back to Cambridge that I realised that the solution was to turn the control unit into a computer in miniature by adding a second matrix to determine the flow of control at the micro level and providing for conditional micro-instructions.”

The Whirlwind was run in a military fashion: the machine was very thoroughly documented and everybody working on the project had to produce a biweekly report. The highly disciplined preventative maintenance programme kept the valve failure rate down to 0.1% per 1000 hours.

By the beginning of 1950 the Whirlwind was running well but the Navy was tiring of the \$1,000,000 annual running budget. However, about that time the USSR developed its atomic bomb and the intercontinental aircraft necessary to threaten US territory. This was also the time of the Korean war and a time of general paranoia in the US. To detect enemy aircraft flying low, a network of small radar stations was required, but this then presented the problem of correlating and presenting a mass of information to the military commanders. As an experiment, Whirlwind was hooked up over Telex lines to a radar in Lexington Massachusetts with some real time computation being performed in the MIT lab. This was a great success, and the airforce took over the project. A new division known as the Lincoln Laboratory was created, and the MIT Digital Computer Lab became part of it. Lincoln's primary responsibility was the Semi-Automatic Ground Air Defence System, otherwise known as SAGE.

Whirlwind ran in this rôle until June 30 1959. One of the project team, Bill Wolf, rented the machine for a dollar a year until the late 1970's after which Ken Olsen, the DEC president, looked after it for a while before transferring it to the Smithsonian.

Probably the most lasting contribution Whirlwind made to mainstream (as opposed to real-time) computing history is the development of the

ferrite core memory, and it is here that the link with DEC becomes clear. Early computer memories, whether of the delay line or storage tube type, were unsatisfactory both because of their bulk and because of unreliability. There is evidence to show that the development of the Williams Tube in the UK gave us a significant technological lead over the US teams, until the arrival of the ferrite core memory. To quote from the famous Moore School course of 1946:

“Several forms of fast internal memory have been proposed and the one that shows the most promise at the present time is the electrostatic storage tube. The one on which most work is being done at the present time by RCA is the Selectron, and when perfected it will have most of the features that are desirable in this type of memory”

The Selectron appears to have been the great white hope of the American designers, but it was a long time coming. It was a complex device, difficult to develop and would probably have been very expensive to mass produce. The Whirlwind used its own electrostatic storage tube, but these designs could not match the elegance of the Williams' tube which was even licensed by IBM for use in the 701. However, all this effort was obsoleted at a stroke by J Forrester's work on ferrite cores. In 1952 the first testable cores were received from General Ceramics, and a  $16 \times 16$  matrix was constructed. To develop a full system required a computer to test it, so the Memory Test Computer (MTC) was built: MTC had the same relationship to ferrite core as the Manchester Mk 1 had to Williams Tubes. The design team was headed by a recent MIT graduate called Ken Olsen. The MTC was therefore the first computer equipped with ferrite core memory. It was a great success, and the memory was transferred onto the Whirlwind, after which the MTBF on memory rose from two hours to two weeks. At this point the tube shop was converted to making displays!

The MTC was about as fast as Whirlwind, although it was much more compact. Since it was not software compatible with Whirlwind, it never became part of the mainstream work of the lab. The designers subsequently moved to the new Lincoln laboratory in Lexington, and Olsen began work on the TX-0 which aimed to test transistor circuitry and a large 64K ferrite memory. The transistors used included the new Philco SBT100 surface barrier transistors which cost \$80 each. TX-0 contained 3,600 transistors.

The TX-0 had an 18-bit data word with a sixteen bit address space and only four instructions: STORE, ADD, JUMP IF LESS THAN and OPERATE. The OPERATE instruction included commands coded on bits that could be combined to produce a large number of sub-instructions such as ‘clear right half of accumulator’ and ‘shift right’. This scheme was to appear in many subsequent DEC computers up to and including the PDP-8. There were two registers: an accumulator and a *live* register used for controlling and buffering I/O transfers. However, there was no interrupt mechanism.

The TX-0 went from Lincoln to MIT in 1958 and was used for teaching and as a laboratory controller. Later the architecture was extended to include index registers. The machine was in use until 1975 when DEC bought it for use in a museum.

TX-0 was a rather futuristic looking machine. Olsen had received some (no doubt harmlessly intended) criticism over the rather homely appearance of the MTC, and seems to have taken this very much to heart in his subsequent systems. He supposedly spent some time in electrical utility shops examining home appliances, and the characteristic toggle switches seen on most DEC computers up until recent times (when toggle switches are frowned upon) are apparently based on those found on 1950’s fridges. Olsen has always been most concerned about the appearance of DEC products and in some parts of the company he is referred to as the Chief Box Designer.

The TX-2 was a much larger machine, containing 22,000 transistors. Its principle design goal was efficient I/O. One option might have been to use a separate I/O processor as with the IBM *channel*. This was rejected in favour of giving the I/O controllers direct access to main memory and having separate program counters with associated program sequences that controlled the I/O using the main processor. It was a short step from here to modern Direct Memory Access using memory mapped peripheral registers and prioritised interrupts, as seen in nearly all subsequent DEC machines. The echoes of this design decision are still very much present today. Intel-style microprocessors follow the IBM I/O model, with separate instructions for performing I/O, in spite of the fact that these microprocessors do not usually have independent I/O processors. Motorola-style devices, including the Rockwell 65xx family, use a very DEC-like arrangement of memory mapped I/O registers.

The TX-2 even had some internal parallelism. It has a 36-bit ALU which could be partitioned into  $4 \times 9$ ,  $2 \times 18$ ,  $1 \times 36$  or even one 9-bit and one 27-bit ALU operating concurrently. The machine had separate adders

for indexing and program counter incrementing. In the 1960's the TX-2 was modified to support multiprogrammed timesharing. It was finally dismantled in 1977.

The construction of the machine was based on a few general purpose circuits mounted in modules running at 5MHz. In 1957, Olsen and two others formed the Digital Equipment Corporation with venture capital from American Research and Development and others. Although the original aim was to build computers, the backers were sceptical and preferred a business based on the construction of logic modules for laboratory use.

For many years, the various series of logic modules were a mainstay of DEC business, but their first computer was shipped in November 1960. DEC machines have always appealed to engineers and scientists because of their low cost and accessibility. The PDP prefix that was used for all computers up to the 1978 VAX machines stands for Programmable Digital Processor, and was used instead of the word 'computer' specifically so that the accountants in DEC's customers' companies would not notice that the engineering teams were buying computers, which as everybody knows are the sole province of commercial Data Processing shops.

Real-time systems, whilst outwardly less exciting than the latest Cray or workstation in fact account for far more of the computers in the world, both by number of units and by shipped value. At a rough count there are fifteen computers in my home, of which only three have keyboards attached. If my TV and audio hardware were a little less aged, there would be far more. So the Whirlwind and its small, fast successors have inherited the Earth, in spite of the scepticism of von Neumann.

---

## Early computers at Manchester University

---

Manchester University is, along with Cambridge University and the National Physical Laboratory, sure of its place in the history books for its pioneering contributions to computer technology. Members of the Society were given some interesting and revealing insights into the stories behind the design of the early Manchester computers at an all-day seminar held at the Science Museum on 23 May 1991. This report concentrates on the hardware aspects of the five machines described.

Professor Tom Kilburn was involved throughout the development period, initially as a graduate student assisting Professor Freddie Williams. He was thus an appropriate choice as the first speaker, and he discussed the development of the both the prototype computer and the full sized successor, the Mark I.

### Mark I

Kilburn told the meeting that the University's involvement with computers started when Freddie Williams paid two visit to the States in 1945 and 1946 to assess the radar circuitry being developed there. On the second trip he visited Bell Labs and saw experiments using cathode ray tubes. The objective was to remove the ground echoes that occurred in all radar systems.

Williams thought he could see a different use for CRTs—as storage devices—and started exploring the possibility as soon as he got back. “We’d been aware of digital computers for some time then, especially about the mercury delay line, but of course the mercury delay line is not immediate access. There was a chance of making an immediate access store, perhaps, with a cathode ray tube. And so it proved.”

After a period of experimentation, a working system was developed, and the need to test it was to prove the foundation of Manchester's computing reputation. “The only way to make sure you’ve got a cathode ray tube store is to actually test it with zeroes and ones changing throughout the pattern at machine speed. Various pieces of test gear were postulated, and Geoff Toothill and I nearly built them, but in the end it turned out that it was going to be far easier to build a computer to do the job properly. We set about that at the end of 1947 and by the middle of 1948 we had

what we called the baby machine.

“The machine attracted a great deal of interest. Among the many eminent visitors who found their way to Manchester was Sir Ben Lockspeiser of Ferranti. His company was already involved in the computer project, as they had supplied a magnetic drum for the baby machine. Now Lockspeiser committed to fund the manufacture of a full scale machine, with the idea of subsequently building copies for sale.

“Following Ben Lockspeiser’s visit, Geoff Toothill and I, aided by Dai Edwards and Tommy Thomas who joined us in September 1948, set about making a big machine. This machine worked. It was quite a sizeable machine and it stayed working at the University for quite a time, and was used by people like Newman, Turing and one or two others.”

It took till November 1949 to complete the specification for the Mark I, and another 15 months before Ferranti was ready to deliver the machine to the University.

Then, said Kilburn, “It took in those days quite a number of months to install the machine. The Mark I was, like all those machines, quite difficult to maintain. We had trouble with the machine, and trouble with the drum, and everyone around us was conscripted to try and keep the machine working. The machine continued to work at the University up to I believe 1959.

“One of our aims was to introduce computers to industry. We allowed industry to use the machine right from the start, charging them some reasonable fee which was I think £50 an hour.”

While grappling with the problems of getting the Mark I to work, Kilburn and his colleagues were already thinking ahead to the next computer.

“It was clear long before the machine actually worked that we could improve on the Mark I. For example the Mark I had been influenced by the pure mathematicians like Newman and Turing that we’d been talking to, so we turned out special instructions to help them. It was clear by 1951 that pure maths would not be the prime use of the machine: scientific computing would.”

That involved designing a floating point accelerator. Together with other improvements suggested by the experience of Mark I, “we embarked on a machine called Meg. This became the Ferranti Mercury, and was about 30 times more powerful than the Mark I at about the same cost.”

## Meg and Mercury

Kilburn was followed to the rostrum by Professor Dai Edwards, who gave the audience many details of the technological advances introduced in Meg and Mercury.

Edwards started by describing the experiences of some of the early Mark I users. One user, which was engaged in working out crystal structures, found that out of the total machine time of 1750 hours, 250 were wasted due to machine faults, about 14% of the time they took. Another 10% of time was daily routine maintenance, so a fair stretch of machine time went to waste. Improving reliability was therefore a priority.

“There were something like 4200 cathodes in the Mark I, but only 1800 in the Mercury with 1600 crystal diodes. Even when you add the totals together there were still fewer elements, a 19% reduction. That was a move towards getting a bit of extra reliability.”

Memory sizes on the Mark I were 10K bits for the CRT main memory, and 650K bits for the drum. These were the same on the successor machines, both Meg and Mercury.

“On Meg we used a CRT, on the Mercury made by Ferranti we went to core. The drum wasn’t that reliable, so on the Mercury it was smaller and more of them could be connected. Thus if one drum went funny you didn’t lose everything. It was possible to attach eight drums, but typically four were used. The IBM 701, a contemporary of our prototype Megs of 1954, had seven times the RAM, but their drums were about 10% smaller.

“We used 10-bit words because that was the natural element that came out of the CRT. With a CRT you have to regenerate the information otherwise it would decay away, so in the Manchester machines there were things called ‘scan periods’ when we regenerated information, and ‘action periods’ when we used the store to get something out or put something in.

“Instructions were 20 bits, so we required two accesses for that, and if we were using a 10-digit number you had a third action. So in three double beats, 60 microseconds, you could do arithmetic on 10 bits, which was done in the B-registers.

“In the Meg we just had five bits to define the instruction, so in modern terminology it was really a risc machine. When we went to the Mercury we used seven bits to provide extra instructions. Short operations were 60 microseconds. A floating point operation with a 40-bit number (a 30-bit fraction and a 10-bit exponent) took 180 microseconds, multiplication 300

microseconds. In the 701 a fixed point addition or subtraction over 36 bits was 60 microseconds, but multiplication was 500 microseconds.”

## **Transistor computer**

Professor Dick Grimsdale followed Professor Edwards with a discussion of his work on the Manchester prototype transistor computer — work which overlapped with the development of Meg/Mercury. This had also emerged out of a concern for reliability, as transistors, though at the time much more unreliable than valves, offered a potential future improvement, and also consumed far less power.

Together with a colleague, Doug Webb, “we were doing some experiments with STC crystal triodes. We got some of the basic circuits, and were able to use them to make a prototype computer. We used a drum because it was the only storage device available. The drum was a delay-type store, with the delay caused by the drum rotation”.

The prototype was a 48-bit word machine, with four spare bits for timing leaving 44 usable bits. The clock rate was 125 KHz. There were 92 point contact transistors in it, with six point-contact diodes on average for each transistor. This machine was first run on 16 November 1953, “the first transistor computer in the world”.

Having proved the feasibility of a transistor computer, “we decided to take it to pieces and rebuild it.

“The full-scale computer had a B-register and an eight word serial register. It also had a multiplier. That machine was operational in 1955. It had 150 watt power consumption and 250 transistors. It would add two 44-bit numbers in 1.5 drum revolutions. The drum operated at 3000 rpm. A division sub-routine took one second, square roots 1.3 seconds.”

History then repeated itself, though: “There was a problem with the unreliability of the transistors”. This was solved by the development of a different type of component, the junction transistor, initially used by Grimsdale in an experimental small core store.

The transistor computer even in its original form had enough potential to interest Metropolitan Vickers. “The outcome was the commercialisation of the machine as the Metrovick 950, of which seven were built. It used junction transistors because they were more reliable.”



## Atlas

The session between lunch and tea was devoted entirely to the advances embodied in the next Manchester computer, Atlas, with further presentations from Messrs Kilburn and Edwards.

Atlas, as Kilburn told the delegates, was a major project for a university to undertake. Value of the first version at delivery was £1.5m, which today is equivalent to £15 million. The later machine cost installed at Chiltern would have cost £25m today. Atlas was competitive with IBM Stretch for the title of most powerful computer in the world in its time, and was 80 times as powerful as Mercury.

“Atlas introduced many new ideas, such as multiprogramming, job scheduling, interrupts, virtual storage, paging and operating systems. The idea was that a job could be input on any teleprinter and output, instead of using dedicated peripherals. The task of sorting it out was given to the supervisor, as no human knew what was going on.”

Its development ensured that in the Flowers report Manchester was designated as one of the three regional university computer centres in the country. Inaugurated on 7 December 1962, Atlas provided a 24 hour service to other universities, including Nottingham, Edinburgh and London, and was operational till 1972, when it was replaced by MU5.

Atlas was produced in a collaborative effort with the university’s long established industrial partner, Ferranti. “We charged 7.5% of the capital value of machine for maintenance”, said Kilburn. “This was £100,000 a year, or £1 million today. We sorted out that the university would have half the machine time, while Ferranti would sell the other half.”

Taking over the rostrum to describe the technical specification of Atlas, Dai Edwards told the audience that Atlas was built from 10 shelves, each with 50 printed circuit boards. In all it contained 60,000 transistors and 300,000 diodes.

Storage used components that were state-of-the-art at the time. There was a fixed store (in today’s terminology, a ROM) of twice 4096 words with 0.3 microsecond access; a core store, with four units of 4096 words and 0.5 microsecond access; and a drum of 100K words.

“In the fixed store were engineering test programs, peripheral start/stop routines, scheduling routines, and 250 additional orders. The fixed store was novel, cheap and fast.”

In 1965 a Dataproducts disc added, 31” in diameter, with 16 or 32

platters on a shaft. The larger size had a capacity of 16m words, or 100Mb. It consumed 7.5 Kw and weighed 3500 lbs.

The processor had a 48-bit floating-point unit, which could be overlapped with a 24-bit fixed point unit. In the multiplier, bits were grouped in threes rather than in twos as on Mercury. Multiply time was 4.7 microseconds.

For I/O Atlas used paper tape equipment, data links over private lines at 60-100 kcps, and also card equipment.

## MU5

Finally, Professor Derrick Morris told the audience about the last of the pioneering Manchester machines. which in the more prosaic computing times of the late sixties was known simply as MU5.

This successor to Atlas was first conceived in 1966. The University won support from both the SRC and ICL in 1967, and had built up a joint design team of 20 by the following year, with 11 from the University, five from ICL and four from the SRC.

The objective of the new machine was to achieve 20 times the throughput of Atlas, of which seven times was to come from improving the technology, two times from the high level language architecture, and two times from the overall system architecture.

Unlike the other Manchester machines, MU5 was conceived as a range of three machines. The first was to be a small machine of around the cost of a DEC PDP-11. The second was to be a high-end scientific machine. The third was to be a multiprocessor. Of these, only the second was actually built.

“The most interesting technical aspect of MU5 was the associative store. This was as a result of an analysis of the Atlas software, especially the instruction code. We learnt something about the frequency of use of operands and control structures. The order code accommodated string functions and vector functions.”

The basic instruction was 16 bits, though there were some 32-bit variants also. There were no conventional registers (such as the block of 128 B-registers on Atlas).

MU5 was heavily pipelined—about five stages at 50ns per stage. The secondary pipeline had 10 instructions. The local store was 250ns, so there was a wide access path to keep system going, with eight instructions in it.

---

## Miscellany

---

- Membership of the Society has risen above the 300 mark.
- Our new chairman, Graham Morris, is the third former BCS president to join the committee. The others are our immediate past chairman, Ewart Willey, and Professor Sandy Douglas. The number will rise to four in October when Roger Johnson takes office as the 1992-93 President.
- Doron Swade is planning to resuscitate the Totalisator Working Party now that his heavy workload commemorating the bicentenary of the birth of Charles Babbage is finished.
- We regret to record the news that one of computing's best-known pioneers, Grace Hopper, died on New Year's Day at the age of 85.

Rear-Admiral Hopper had many innovative software developments to her name, though perhaps she is best known for introducing the word "bug" into computer terminology, when she discovered that a failure of the Harvard Mark I computer was caused by a moth.

- Computing anniversaries in 1992 include:
  - 375th: Napier's invention of the primitive calculating device known as Napier's Bones in 1617
  - 100th: Herman Hollerith's entry into business as the Hollerith Electric Tabulating System in 1892, following the success of his automation of the processing of the 1890 US census
  - 30th: launch of the Ferranti Orion in 1962
  - 25th: IBM's introduction of the binary synchronous communications protocol in 1967
  - 25th: installation of the first cash dispenser in the UK, also in 1967
  - 10th: the issuing of a licence to Mercury Communications in 1982
  - 5th: launch of the IBM PS/2 in 1987.

---

## Working Party Reports

---

### **Elliottt 803 Working Party**

*John Sinclair, Chairman*

In the last issue of *Resurrection* I reported that the processor was operational. Since then, we have made steady progress towards our goal of restoring it to “as new” condition, as a result of a lot of hard if unglamorous work. The working party members put in 32 sessions of restoration during 1991.

The processor is now exceptionally reliable — indeed, I cannot remember the last logic failure. We did have a store problem in the summer, when the room temperature was around 80 degrees (neither the room nor the processor has any air conditioning).

All the peripherals are now operational, including the 35mm film handlers, and reliability is steadily improving. The film handlers were thick with grime when we got them, and to start with we were discovering new faults every time we switched them on, but we have now got past that point.

The major advance since the last newsletter has been the acquisition of a new battery, generously donated by the RAF to the Science Museum after we tracked down the appropriate stores unit and paid them a visit. (The batteries are still being made for use in aircraft like the Nimrod.)

This was the first time that any member of the working party had seen a brand new battery: the normal procedure with defective batteries when the Elliotts were in use was to replace the faulty cell or cells. This was principally on grounds of cost — the last time I looked each cell cost around £70.

The new battery has helped a great deal. Previously, we had to put the old battery on a charge for an hour before we could switch the processor on, as it would not otherwise cope with the surge of power.

Other acquisitions include a collection of Creed teleprinter parts, acquired by working party member George Bradley from an engineer who used to work with Creed, and had been storing them in his garage since the company discontinued production. These will be useful for the Pegasus as well, and we should now be able to keep the teleprinters running on both machines for a considerable time.

At the Open Day, we were promised a complete additional 803 — a

major surprise, as we thought that our machine plus the incomplete one in the Science Museum store were the only surviving examples.

## DEC

*Adrian Johnstone, Chairman*

The Working Party, having restored the museum's original PDP-8 to its full glory, has been concentrating its efforts on a PDP-12. This machine is rather a surprise to anyone that associates DEC with small machines, firstly in that it is rather large, and secondly in that it is green. I cannot, off hand, think of any other green production computers (I discount the Elliot 401 languishing in the corner as it was never mass produced) and am sure that the particular combinations of day-glo and olive drab employed here will not be seen again in our lifetimes.

The 12 is an odd machine in many other ways. It is something of a specialist device, being equipped as standard with a bay of Analogue to Digital Converters and a console VDU. Internally, it is a hybrid of a PDP-8 and a LINC (Laboratory Instrument Computer) which was originally demonstrated at MIT in 1961. The LINC was a great influence on the design of the DEC PDP-4 and PDP-5, and therefore of the 5's successor the PDP-8. The original LINC machines were constructed using DEC supplied logic modules, so it was natural for DEC to take over production from MIT. Subsequently a two processor machine, the LINC-8 was designed which could execute LINC and PDP-8 instructions in parallel, and then the PDP-12 which could execute either LINC or -8 instructions but not at the same time.

About 1000 PDP-12's were made, and many were sold to hospitals and other medical institutions. Our machine was used for diagnosis and research into hearing and speech disorders. A speech synthesiser was used to play back syllables to the guinea pig in a controlled way. This system, complete with software is now restored, and can make various burbling and whistling noises. The machine was never used for full connected speech, so it cannot easily be persuaded to hold a conversation, but we plan to use some public domain text-to-phoneme software to build some sample sentences for play back. This is a major undertaking, and I would be pleased to hear from programmers with an interest in speech synthesis. I have built a small phoneme based speech synthesiser for use on my own machines on which the text-to-speech programs have been demonstrated, so most of the ingredients are already available.

The Working party has also acquired some PDP-11 equipment which will shortly be transferred to the museum for restoration. The haul includes what is probably one of the first PDP-11's to be sold in this country. Unfortunately the machine is showing signs of *rust* of all things, so considerable work may be required before it may be safely switched on.

## **Software and Emulators**

*Tony Sale, Chairman*

The Working Party took the major decision last summer to focus its efforts more sharply by concentrating on the emulation of old computers, and particularly the computers in the Society's collection. Software in the round had proved to be too large a subject.

As a result, we are now known as the Software and Emulators Working Party. At the same time, I succeeded Martin Campbell-Kelly as chairman.

We have made considerable progress since then. We have spent some time studying the possibility of developing standard file formats for paper tape emulation on floppy discs — for transferring data and programs from tape to disc for use in emulations and vice versa. We have developed a firm proposal which we are now considering in more detail.

We have also spent some enjoyable hours testing Chris Burton's "flight simulator" of Pegasus. Some explanation is needed here, as the flight simulator concept is an important one which will be of interest to everyone attempting conservation and restoration of historic computers.

The "flight simulator" is a piece of software that not only emulates the instruction set of the target computer, but also emulates the person-machine interaction using graphics. You could say it conserves the persona of the machine in software.

Thus Chris Burton's simulator shows a pictorial representation of the front face of a Pegasus with its CRTs, lights and switches working in real time when a program is run. (A hand emerges from behind the screen to change the switch settings when this is required!) It also shows the holes punched in the paper tape that is being used.

Chris' simulator has now been extensively tested and debugged, and has been verified by Derek Milledge against our own Pegasus. Some of our older members are now using it to redevelop programs they wrote when Pegasus was a new machine.

A second flight simulator is being written by Peter Onion for our Elliott

803, and this is now nearing completion. This, like the Pegasus program, will run on any PC that has VGA graphics.

Writing simulators is as you can imagine very time-consuming. Another thing we have been looking at is the development of a toolset to enable these programs to be written more easily.

## **Pegasus**

*John Cooper, Chairman*

In my last report I described how we had succeeded in getting Pegasus working again. Getting from that point to a condition where the system would be reliable enough for continuous daily use has involved at least as much work again.

There has been a sustained effort over the past year to improve reliability, involving 34 sessions of work by Working Party members. The major achievement has been the complete refurbishment of the margin control panel.

This is a panel inside the power supply unit that allows you to change the voltages within the machine gradually until a component within the computer fails, and then to identify and rectify the cause of failure. Getting it into full working order was a major job. The panel will help us greatly in our goal of restoring Pegasus to routine operational condition, and we are now embarking on a programme of work designed to achieve this.

We are keen that this work should have an educational as well as a functional value, and are actively encouraging younger members of the Society to join us so they can familiarise themselves with the logic and operation of Pegasus. Anyone interested should contact myself or Tony Sale and we'll arrange a meeting. No previous experience of Pegasus or any similar computer is required.

We also had a problem with the drum, which we identified using the margin control system though it proved not to be related to voltage changes. It took us a long time to track down the fault, which turned out to be a bad joint inside the drum case.

We have completely refurbished the package tester, which also had a major fault. This is an aid to repairing broken packages — circuit boards — and was supplied with the system.

Away from the machine, we have made further progress in cataloguing our collection of spares. We are also in the process of cataloguing the

software, and copying it so that we have duplicate tapes of everything.

Chris Burton has produced a Pegasus emulator which runs on an IBM style computer with VGA graphics. It emulates both the functions and operation of the machine, and presents a most realistic view of the console in colour, with a full set of working handswitches that can be manipulated via the IBM keyboard. This is a very fine piece of work.

We also now have available a paper tape editing set for use on an IBM PC. This allows the user to create programs and text using a standard editor or word processor, with conversions to and from five hole Pegasus tape being performed automatically.

## **S-100 bus**

*Robin Shirley, Chairman*

Interest in the activities of the S-100 group has steadily been increasing. A lot of people have been contacting us after a news item about our activities appeared in Computer Shopper magazine. We also had a much more productive Open Day than in 1990, which apart from providing us with some interesting contacts also produced some bits and pieces of useful equipment. Membership of the Working party is now between 10 and 20.

We have acquired a Dynabyte machine, complete with documentation, which was on view at the Open day. Our display then was similar to the 1990 show. I am currently exploring the possibility of acquiring at least one Altair, which is of particular interest to us as it was the original S-100 bus machine.

Many of the people who have contacted us have specific interests. For example, one man based in Carlisle has a collection of Triumph-Adler machines—he ran the Triumph-Adler user group for a while. Another has a good collection of Hewlett-Packard desktops. A third used to run the Lynx User Group.

This leads to the thought that we should perhaps create a number of sub-working parties to cater for each of these interests, all linked to the parent group in a tree-like structure. Our working party does not have the same obvious focal point of interest as the Pegasus and Elliot groups. I am contemplating writing a news sheet to keep all our members informed of the many different interests within the Working Party, if this is felt to be worthwhile.



---

## Forthcoming events

---

**5 October 1992** In steam day

**29 October 1992** Evening meeting

**2 November 1992** In steam day

**19 November 1992** Society Open Day

**26 November 1992** Evening meeting

**7 December 1992** In steam day

In Steam Days start at 10 am and finish at 5 p.m. Members are requested to let the secretary know before coming, particularly if bringing visitors. Contact him on 071-938 8196.

Members will be notified about the contents of the evening meetings once the Committee has finalised the 1992-93 programme. All the evening meetings take place in the Science Museum Lecture Theatre and start at 5.30pm.

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

Editor – Nicholas Enticknap

Cover design – Tony Sale

Typesetting – Adrian Johnstone

Printed by the British Computer Society

©Computer Conservation Society

---

## Committee of the Society

---

*Chairman* **Graham Morris FBCS**, 43 Pewley Hill, Guildford, Surrey GU1 3SW. Tel 0483 66933

*Secretary* **Tony Sale FBCS** Manager, Historic Machines Programme, The Science Museum, Exhibition Road, London SW7 2DD. Tel: 071-938 8196

*Treasurer* **Dan Hayton**, 31 High Street, Farnborough Village, Orpington, Kent BR6 7BQ. Tel: 0689 52186

*Science Museum representative* **Doron Swade**, Curator of Computing, The Science Museum, Exhibition Road, London SW7 2DD. Tel: 071-938 8106

*Chairman, Pegasus Working Party* **John Cooper MBCS**, 4 Tower Road, Belvedere Road, Kent DA17 6HX. Tel: 071-477 8000

*Chairman, Elliott 803 Working Party* **John Sinclair**, 9 Plummers Lane, Haynes, Bedford MK45 3PL. Tel: 02306 6403

*Chairman, DEC Working Party* **Dr Adrian Johnstone CEng, MIEE, MBCS**, Department of Computer Science, Royal Holloway and Bedford New College, University of London, Egham, Surrey TW20 0EX. Tel: 0784 443425

*Chairman, S100 bus Working Party* **Robin Shirley**, 41 Guildford Park Avenue, Guildford, Surrey GU2 5NL. Tel: 0483 300800

*Editor, Resurrection* **Nicholas Enticknap**, 4 Thornton Court, Grand Drive, Raynes Park SW20 9HJ. Tel: 081-540 5952

*Archivist* **Harold Gearing**, 14 Craft Way, Steeple Morden, Royston, Herts SG8 0PF. Tel 0763 852567

**Dr Martin Campbell-Kelly**, Department of Computer Science, University of Warwick, Coventry CV4 7AL. Tel: 0203 523196

**George Davis CEng FBCS**, 25 Manor Way, Purley, Surrey CR8 3BL. Tel: 081- 660 5581

**Professor Sandy Douglas CBE FBCS**, 9 Woodside Avenue, Walton-on-Thames, Surrey KT12 5LQ. Tel: 0932 224923

**Christopher Hipwell**, Reed Business Publishing, Quadrant House, The Quadrant, Sutton, Surrey SM2 SAS. Tel: 081-652 3149

**Dr Roger Johnson FBCS** 9 Stanhope Way, Riverhead, Sevenoaks, Kent TN13 2DZ. Tel: 071-631 6388

**Ewart Willey FBCS** 4 Sebastian Avenue, Shenfield, Brentwood, Essex CM15 8PN. Tel: 0277 210127

**Pat Woodroffe**, 9A Guildown Road, Guildford, Surrey, GU2 5EW. Tel: 0483 63299