Further developments

The LITHP implementation is part of a wider project to offer list processing facilities to users of the I.C.T. 1905, irrespective of the language in which they wish to program.

The LITHP system enables list processing programs to be written within the ALGOL system. We are indebted to the Atlas Computer Laboratory at Chilton for permission to implement the SLIP system on the I.C.T. 1905, thus allowing list processing programs to be written within the FORTRAN system. At the same time work is progressing on a machine-code list processing system, and on a CPL compiler which will incorporate list processing.

Correspondence

Multiple precision arithmetic (real and complex)

To the Editor,
The Computer Journal.

Sir,

We should like to draw attention to the facilities mentioned at the end of the paper, "The main features of Atlas Autocode" (published in this *Journal*, January 1966, p. 303). These provide an extensive range of arithmetical operations on real and complex numbers of varying precision (single, double and multiple). These operations include ± * / and the elementary algebraic and transcendental functions. They have been in use for some time now and can be regarded as fairly extensively tested. (One of the test programs calculated π to 5000 decimals.) We feel that they may be particularly useful because they are part of the language system which is also used for the more conventional parts of the calculation.

Examples of the special arithmetic statements are:

(i) **mr** (I) a, b, c
will declare a, b and c as multi-length real numbers of length I, I being a conventional integer expression.

(ii) **dc array** H $(1 : n)$
will declare H as a n-element vector of double length complex numbers.
The data evaluation statement is of the form

$$[\mathrm{A}1\ \alpha1\ \beta1\ \mathrm{A}2\ \beta2\ \mathrm{A}3 \ldots]$$

where the A's are operands, the α's are unary operators and the β's binary operators ("unary" meaning one operand, and "binary" two).

The above example means fetch A1, perform the operation $\alpha1$, then perform $\beta1$A2 on the result, and then $\beta2$A3 on that result, and so on. The entire facilities are contained in the operators α and β and their effect on the different types of operand.

A document entitled "Notes on the Special Arithmetic Statements in Compiler ABC" which describes in detail the new facility is available upon request. It also gives performance figures.

Yours faithfully,
S. R. Clark
W. F. Lunnon

Department of Computer Science,
The University,
Manchester 13.
9 May 1966.

Progress?

To the Editor,
The Computer Journal.

Sir,

I write to comment on two items in the January 1966 issue of the *Journal*. In both cases I am forced to wonder if we have learnt anything about computers in the last ten years or so.

First, the apparently magical properties of relocatable binary, as described in the article on the 'Egdon System'. Are we seriously to believe that in 1966 relative addressing is being presented as a new discovery? To mention only one machine, the whole of the Pegasus assembly system uses just this method of relative addressing in binary and in machine code, off paper tape and magnetic tape, and has done so since 1956, when the machine was first available. Incidentally, Stanley Gill told me recently that he had always regarded the Pegasus system as a distinct step *backwards* from EDSAC 1! In some ways the remarkable thing about the Egdon system is that there are people who think it is remarkable.

Secondly, the advantages of magnetic tape with fixed size, pre-addressed blocks, as used on Atlas. Except for the size of block and that the computer can be used as a tape-addressing machine, this is the tape system that has been in use on Pegasus machines since about 1958 (and it was used on EDSAC 1 well before then!). It is, of course, too late to reverse the mistaken decision (?) of the early 1950's to have completely variable block length, now apparently widely adopted on grounds of 'compatibility', without any regard as to whether this system is the one of most use to the customer.

Yours faithfully,
Paul A. Samet

Director, Computation Laboratory,
The University,
Southampton.
10 March 1966