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) \[ \text{mr (I) } a, b, c \]
will declare \( a, b \) and \( c \) as multi-length real numbers of length \( I \), \( I \) being a conventional integer expression.

(ii) \[ \text{de array } H (1 : n) \]
will declare \( H \) as a \( n \)-element vector of double length complex numbers.

The data evaluation statement is of the form

\[ [A1 \alpha \beta1 A2 \beta2 A3 \ldots] \]

where the A’s are operands, the \( \alpha \)’s are unary operators and the \( \beta \)’s binary operators (‘unary’ meaning one operand, and ‘binary’ two).

The above example means fetch \( A1 \), perform the operation \( \alpha1 \beta1 A2 \) on the result, and then \( \beta2 A3 \) on that result, and so on. The entire facilities are contained in the operators \( \alpha \) and \( \beta \) 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.