

due to the general nature of the routines, but have not yet had time to establish the full extent of this. It is estimated that the average is of the order of 20%.

Development times for projects are longer than had been hoped, and FACT is very heavy on machine time. There are exceptions; some programs have been developed very quickly, but during the learning period progress has been discouraging.

We are very concerned about the cost of making even minor alterations to a program after it is working. Even in the best regulated circles minor systems changes are necessary, and in a dynamic organization where improvements are constantly being sought, changes could well be frequent. FACT enables the changes to be made easily and in a comparatively short elapsed

time, but the cost in machine time can be considerable. This is a problem we shall soon have to face.

In conclusion, although we have had a very trying time since last October, we have got over the learning period and are convinced that FACT is a most powerful language, and the debugging facilities are excellent.

We now know the weaknesses and the pitfalls to be avoided and shall plan future jobs accordingly.

We also intend to set up a small team to investigate the unsatisfactory features with a view to improving the efficiency of the object programs.

One point may be of interest, whenever anything goes wrong the first reaction is that it is either the program or the operator at fault—never the machine or the software.

Book review: Computer organization

Workshop on Computer Organization, Edited by ALAN A. BARNUM and MORRIS A. KNAPP, 1964; 242 pages. (London: Cleaver-Hume Press Ltd., 72s.)

This book is largely concerned with the unconventional for its own sake. Eight papers and subsequent discussions have been reproduced from a conference held in October 1962 in Baltimore, Md.

Two main types of machine organization interest the participants. One is the array of active computing elements, each with its own data store, obeying the same instruction stream from a single control unit. This is exemplified by the SOLOMON computer in Chapter 2 (D. L. Slotnick *et al.*), of which a fuller account will be found in *Proceedings of the F.J.C.C.* 1962; also (in Chapter 3) by a form of matrix organization using optical logic with which readers of *The Computer Journal*, July 1963, will be familiar. The main application is to the sort of numerical problem which is naturally represented on a (rectangular) grid of points, at each of which virtually the same calculation has to be performed, using numbers associated with the point, with its neighbours, or with a "common" pool of parameters. There are two sorts of "edge effects" which arise: one from those points representing the physical boundaries of the problem, the other from the limits of the computing grid, on which the problem is represented by successive overlays. To handle such special cases a form of modal control is introduced, which allows the activity of certain elements to be suspended while others follow through a particular instruction sequence. Obviously the fewer "computer edges" the better. SOLOMON has 1,024 computing elements, and on a selection of suitable problems a performance gain of around 100:1 is claimed in comparison with a "conventional computer." It can also be seen that this organization may be applied to any calculation which has to be repeated many times with slowly changing parameters—slowly changing in the sense that not too many elements will be held up waiting for the last one to go a few more times round a loop. Numerical problems can conceivably be found to satisfy this requirement, and a computer system with a modest number of elements built to process them economically. It is interesting to enquire whether similar situations could be found in data-processing work; on the whole the chances of doing so seem slimmer—the transition between "M" and "F," for example, is usually taken rather abruptly.

Thus the second form of organization requires a multiplicity of control elements, each following its own sequence of instructions, sharing access to data in such a way, one hopes, that an effectively determined calculation is performed. The model taken here for such systems is the Iterative Circuit Computer or Holland machine (see the articles by J. H. Holland in *Proceedings of the E.J.C.C.* 1959 and *Proceedings of the W.J.C.C.* 1960 for up-to-date accounts of this class of machines). I am not sure whether they fulfilled Holland's original purpose, but as starting points in computer design they are rather like skyscrapers without lifts, corridors or telephone systems, taken as starting points for office accommodation. Two of the chapters (4 and 5) in the book and much of the discussion are taken up with, yet nearly miss, this point. When the (blue) sky's the limit there is surely no harm in taking the few steps necessary to get a sensibly programmable system.

The remaining papers offer adjuncts to mainstream programming rather than radical departures. Chapter 1 (D. P. Adams) suggests a mechanization of nomograms, with countable bits on a magnetic or photographic recording replacing the numerical scale markings. It is not clear why this is being attempted, since it appears to offer no advantage over conventional approximation and interpolation techniques. Chapter 6 (G. Estrin *et al.*) describes briefly the efforts being made at U.C.L.A. to devise systematic ways of enhancing the performance of a conventional computer by attaching special-purpose computing elements to work in parallel with it. Chapter 7 (G. J. Culler) outlines a Man-Machine Communication system which sets the fingers twitching, if not the brain. Finally, the paper by N. S. Prywes and S. Litwin describes a design for a computer capable of processing data organized in a generalized form of list structure.

On the whole, therefore, a very disappointing contribution to this interesting field. At least some of the participants seem to have felt the same way, and Professor Pasta leads in with some politely searching questions. It is a little surprising to find exactly the same searching 100-word question on p. 173 as on p. 149, and the verbatim style only heaps on the confusion. There are many errors which one would have expected the editors to notice, but I think the prize goes to the peripatetic (yet powerful!) "touring machine" which appears briefly on p. 211.

J. K. ILIFFE.