Again, caution should be exercised in determining the numbers of commands necessary to perform these operations so that a generalization of the procedure will not entail too extensive housekeeping.

The first of these logical problems may be solved by a number of different approaches. However, the second problem, the interpretation of the keyword for unpacking, requires an instruction which examines the word digit by digit and controls the storing of the significant components in their proper position in the vector, suggesting a type of conditional jump command. The zero suppression multiply may be accomplished by forming the logical product of the keywords of two vectors, forming a new keyword which would control the multiply sequence exactly as the unpacking procedure. To illustrate the advantage of this operation, let it be supposed there is the vector,

$$\mathbf{y} = (2, 1, 0, 4, 0, 0) \qquad (4)$$

with the binary keyword,

$$110100 \qquad (5)$$

and suppose forming the scalar product of **y** and the vector **x** given by equation (1) is desired. This would be accomplished by forming the logical product of the keywords (2) and (5)

$$\begin{array}{ll} (2) & 101001 \\ (5) & \underline{110100} \\ & 100000 \end{array} \qquad (6)$$

indicating that the only corresponding significant components of the two vectors are the first, and the scalar product would then involve only a single multiplication rather than six multiplications and five additions.

It is estimated that a computer designed with all of the afore-mentioned qualities would reduce the running time of present procedures by as much as a factor of ten. Since most of the requirements of this problem are compatible with many other types of problems, and furthermore, since many of these features may be found on existing machines, such as the Univac Scientific, it is the contention in this paper that all of these requirements should be fully investigated so that they may be incorporated in the future machines of this industry.

Considerations in Making a Data-Gathering System Compatible

B. L. WADDELL

THE paper discusses the design problems facing the data systems engineers who are required to produce a data-collection system that will be able to enter a computer easily. Some empirical formulas are presented with a discussion of how to use these formulas. The many recording tools and their application to data systems which are being prepared for entry into computers are described with a discussion of the place of each recording device. The second section of the paper is a critical analysis of four data recording or gathering systems designed to go directly to a digital computer.

B. L. WADDELL is with G. M. Gianninni and Company, Inc., Pasadena, Calif.

Using a Variable-Word-Length Computer for Scientific Calculation

FRED GRUENBERGER   E. H. COUGHRAN

IN discussing the use of a variable-word-length computer, this paper will be restricted entirely to past history; that is, ideas and practices that are actually in operation. This implies, of course, that only the *702* will be talked about and the implied comparison to fixed-word-length machines is to a machine like the *701*.

FRED GRUENBERGER is with the General Electric Company, Richland, Wash., and E. H. COUGHRAN is with the International Business Machines Corporation, Richland, Wash.

At Hanford, Wash., the *702* has been used for scientific computing, with what is regarded as considerable success, since its installation in June 1955. The percentage of available machine time devoted to numerical analysis has steadily increased, standing currently at about 20 per cent. This is not to say that mathematics is taking time away from commercial work, but rather reflects the increase in efficiency on commercial problems and a general increase in mathematical problems as time goes by. Scientific computing has covered a wide gamut of problems, including linear programming, numerical integration, differential equations, and many complicated formula evaluations, as well as work on reactor data and weather data which might be called scientific data processing.

This work demands one large machine to serve for both commercial and scientific problems. This is a common situation, and perhaps any machine would suffice; it has been pointed out that through an automatic programming system, any machine can be made to appear like any other. Indeed, most programming is done with reference to the automatic programming system rather than the particular machine. The main contention here is that a variable-word-length machine makes the over-all job easier and, in particular, it is felt that the variable-word-length machine offers some distinct advantages for scientific computing.

First, since the word length is arbitrary, floating decimal or fixed decimal calculations can be carried to any degree of precision with equal efficiency and equal ease of programming. At present, the floating routines use a 10-digit mantissa with a 2-digit exponent, which is treated as a single 12-character word; the exponent has a range of 200. Should greater (or less) precision be needed, no great effort would have to be expended to alter the routines for any other word length; in fact, the word length can be made a parameter. The appearance of the machine to the programmer is the same in all cases, and programs written for one word length will function properly in all cases, both on paper and in machine code. Moreover, the variable-word-length machine makes it attractive to carry through many calculations completely in fixed point which would certainly call for floating point on a fixed-word-length machine. It is awkward to do multiple-word arithmetic on a fixed-word-length machine, but easy to do the equivalently precise work on a variable-word-length machine. A fixed-word-length machine working in its own word length is obviously at an advantage; it is when the work calls for an odd word length that the variable-word-length machine excels.

Generally speaking, for an installation doing only scientific work, it would be natural to think of a pure computer, such as the *701*, *650*, or Datatron. However, even in this situation, the variable-word-length machine offers an advantage, if the data to be handled are intrinsically of an odd length. In data reduction, for example, large volumes of 2- or 3-digit numbers tend to waste either memory space or machine time or both, in a fixed-word-length machine.

There is much to be gained from having two groups of people working around one machine within one organization. The diverse backgrounds and experience of commercial and scientific people blend into a powerful team when the common problem, outwitting a machine, is approached. Programming insights and triumphs of either group become helpful to the other. For example, the floating point arithmetic subroutines, developed for the numerical analysts, have been used on commercial problems, where a given variable, such as stock quantity, may have low precision but a wide range. Similarly, a routine of particular value to the commercial people for producing tabular reports has proved to be just the thing for processing meteorological data and preparing tabular reports. A generalized sorting routine has become widely used by both groups, since internal sorting is common to all sorts of problems. Characteristically, commercial people have a background which is strong on procedures and flow-charting; mathematicians have an ingrained love of precise definition and compact symbolism. The two groups thus tend to complement each other.

It seems to be intrinsic to a data-processing machine like the *702* that it become surrounded by a greater quantity of more flexible peripheral equipment than does a pure scientific machine. This is no drawback to doing scientific computing on such a machine. For example, there was an initial feeling at Hanford that problems from the numerical analysis unit involving only a small amount of the peripheral equipment could be sandwiched in between commercial problems which tend to use every gadget available. This has not come about. The mathematical programmers have used the devices available to the fullest, with a corresponding speed and efficiency increase.

It is reasonable to assume that, regardless of the nominal capacity of main memory for a given machine, one pays for memory in direct proportion to the number of characters one can store at one time. Here the variable-word-length machine pays off again; it is parsimonious of main memory and tape memory, and hence of tape-read time. Main memory is always packed solid, regardless of how various numbers vary in length. Extremely long (blocked) records can be recorded on tape so that tape is packed with information and very little tape movement is wasted. It should be recognized that the *702* is an alphabetic machine, so that for pure numeric work some of its memory capacity is perforce wasted. Oddly enough, the drum on the *702* is very nearly a fixed-word-length device, with the word length 200 characters; precisely for this reason, drug storage space is sometimes wasted.

It is felt that a variable-word-length machine helps to ease the burden of devising an automatic programming system. This is difficult to prove or disprove; it can only be pointed out that this one was written and "debugged" in a matter of a few months. What seems clearer is that changes in the system are relatively easy to make on such a machine, and are being made constantly on the basis of developments and new requirements.

Under some conditions matrix algebra is at an advantage on a variable-word-length machine. In a large linear programming problem, it was found that a scalar product using a modified 10-digit floating arithmetic took roughly the same amount of computation time as the same scalar product using a fixed 15-digit word arithmetic.

Due to the serial nature of the *702*, built-in automatic checks are cheap and plentiful; no machine time need be wasted in programming the checks. Incidentally, the record of the *702* for not passing undetected memory errors is partically perfect.

An interesting feature of the *702* is that the machine seems to get better, programming-wise, as time goes by. On other equipment, the weaknesses, and features which should have been built in but were not, rapidly become of great importance to the programmer. Naturally, machines of infinite speed and capacity, taking no space, and renting for $5 a month would be liked by all; short of that, the designers of the *702* deserve commendation for a splendid piece of equipment. If one were to talk to the designer of a computer, one would probably ask "Why didn't you?" and he would always have a ready answer. When, however, one asks "Why did you?" he might be quite embarrassed; the point is, few "Why did you?" type of questions have been found for the *702*.

One of the significant advantages of the *702* accrues from its very simple structure. This is evidenced in both scientific- and commercial-type problems in two ways: (1) The simple structure of the machine makes it possible to define an abstract or "automatic coding" system from without rather than from within. That is, the machine is sufficiently simple and yet sufficiently flexible so that any automatic coding technique or philosophy can be described from the point of view of the problems to be solved and the demands for particular manipulations inherent in those problems, rather than starting from a set of machine restrictions and doing the best one can or going to great length to overcome inherent difficulties. (2) The machine is extremely easy to code in its own language, if necessary (indeed, where the number of different jobs is small, this is the only method used at some installations). This feature gives one tremendous power in two operationally vital areas. First, in processing a job, when machine failure, operator error in judgement or accident, or erroneous or unanticipated data cause one to "fall flat on his face," it is not necessary to "go back to the drawing board." It is possible, and, on the basis of this experience, eminently practical, to pick oneself up off the floor. Several times, in one or the other of these circumstances, a corrective set of instructions has been programmed, keypunched, and a set of correct answers run off within the hour. Second, at certain unpredictable and unannounced times, problems of moderate difficulty are presented to the computing section for immediate solution. These are problems of extremely high priority with a machine solution time of 1/2 to 3 hours and a total elapsed time from problem presentation to final solution of 4 to 24 hours. This time scale does not permit the normal cycle of algebraic manipulation, programming, debugging, test cases, and solution. The *702* is simple enough in its order structure to permit coding on an "on the spot" basis. Note particularly that corrections or other machine instructions can be entered directly via the card reader or console.

The last-mentioned advantage is part of the final point: The *702* is decimal and alphabetic, which makes for ease of communication between the machine and the programmers, and the machine and the console operators. Further, although numerical analysts work largely with a 10-character vocabulary, the fact that the *702* has 47 characters available is frequently useful, if only for dressing up printed reports.

Lest it be thought that the picture is too rosy, let some of the disadvantages to using a variable-word-length machine for scientific computing be stated. (1) The chief disadvantage probably lies in the economics of the operation: A pure numeric, fixed word-length, parallel machine with equivalent circuit speeds would perform each operation at greater speed, hence lower cost, particularly on long production runs where the word length happened to fit the problem's needs or could be made to fit. This statement would apply with even greater force to a machine with built-in floating point. (2) For some types of logical mathematical work, there is an advantage to using a binary machine, and all present variable-word-length machines are decimal. (3) The variable-word-length principle itself is responsible for programming slips that might not otherwise be made, in that word boundaries occasionally slip. For example, it is easy to store a 4-digit word at a place in memory where a 3-digit bucket has been set up; this defaces the word to the left. It is hoped that much of this sort of nuisance will be eliminated through improvements in the next automatic programming system.

The dual-purpose installation has only one serious drawback, which comes about from the diametrically opposite nature of the work of the two groups when they are sharing one machine. By and large, commercial problems tend toward the routine, "chiseled-in-granite," repeating problems; numerical analysis problems tend to be on-demand one-time (or few-time) affairs. This can lead to questions of priority; fortunately for this thesis, there are questions of priority with any kind or kinds of work on any type of machine.

On the whole, however, it is felt that the advantages far outweigh these somewhat tenuous disadvantages. Attention should be focused on the possibility of adding numerical analysis work to a commercial installation and/or the attractiveness of a dual-purpose installation. In any installation involving commercial problems, the record-keeping (which is the essence of present-day commercial work) gets done; there may then be an opportunity to work on commercial problem solving. When that situation evolves, it may be quite a help to have a scientific group at hand. Under a common leader, the two groups become compatible and ideas cross-fertilize each other.

Unusual Problems and Their Solutions by Digital Computer Techniques

L. ROSENFELD

TEN years ago the primary role of an electronic computer was to develop mathematical tables and to solve the scientific problems which arose during the research and development work being done at that time for the military and other government agencies. Since then, and particularly of late, the emergence of the computer as a powerful and versatile tool for the management of most major business and industrial corporations in assisting them to carry out their normal business operations has been witnessed. These would include such standard operations as payroll accounting, inventory control, production scheduling, invoicing, billing, and cost accounting.

It is natural that company management should now seek other fruitful areas where the computer might be utilized for profitable advantage. By and large, with the possible exception of those companies which are large enough to afford to have on their staff a group similar to an operations research group, the places where management will seek additional areas of work for its computing facility will be chiefly limited to those areas which are normally under the cognizance of the comptroller or finance officer. Usually, management will not think in terms of using the computer as a device which could possibly aid in improving a company system or subsystem which is part of the way by which part or all of the business of the company is carried out. For example, management of a large trucking company probably would not use the computer to help determine the most economical time to replace worn-out vehicles though it might use it to bill its customers; similarly an automobile manufacturer probably would not use the computer to predict what his spare parts inventory should be for the next 10 or 20 years although he would use it to maintain his inventory.

This paper is essentially a brief description of three case histories of what would generally be defined as operations research

L. ROSENFELD is with Melpar, Inc., Boston, Mass.