Useful Applications of a Magnetic-Drum Computer

STANLEY FRANKEL

The general applications and the operation of magnetic-drum digital computers are described by use of a detailed explanation of the LGP-30, a typical example of this class of computers.

ONE of the most useful and flexible types of computer in current use is the magnetic-drum digital computer. The range of suitable applications for this type of computer is explored in this article. The Librascope, Inc., type *LGP-30* computer (Fig. 1), here used as a typical example of magnetic-drum computers,[1] is described in sufficient detail to explain its elementary operations and to demonstrate how these elementary operations can be combined to perform complex computations. It is not possible to present a precise definition of the class of problems to which these machines can usefully be applied. Instead, these problems will be sketched in broad outline, as a part of the more clearly definable class of problems which can, in principle, be solved by machine computers.

In approaching the problem of useful applications of drum computers, we first ask the broader question, "What tasks can be performed by any digital computer, without regard for speed, convenience, or economy?" It is convenient to consider together the human computer (equipped for example, with a standard desk calculator) and the machine digital computers. We thus ask, in brief, what is a *computation*? Computations of the most familiar type are straightforward sequences of arithmetic operations. An example is the evaluation of the entries in a mathematical table. It should be noted, however, that neither the mathematical analysis leading to a formula for the function nor the printing of the final volume is included in the customary meaning of *computation.*

There are other operations, also called computations, which are not so clearly of arithmetic character. One example is the process of sorting a set of words into alphabetic order. This example still has a somewhat arithmetic character since the alphabetic ordering of words can easily be represented by a *greater than–less than* relationship of numbers. A more clearly nonarithmetic example of computation is the translation of prose from one language to another. It now seems to be possible to perform this task usefully, if not with great literary merit, by machine.

The common feature of these computations is that a body of information is somehow transformed or manipulated. The output resulting from a computer's activity is information, numerical or otherwise. Similarly, the computer requires an input of information which is transformed into the desired output. Not all of this input information is numerical in the ordinary sense; another part is a set of instructions which control the activities of the computer. There may not be a sharp distinction between these two types of input.

Often instructions are presented to a computing machine in a numerical code so as to standardize the way in which it receives and holds information. In the early digital computers, and in some designed for special purposes, much of the instructional information is set into the machine by wiring, either permanent or replaceable. In the modern "general purpose" computers most of the instructions are introduced in numerical guise and look much the same as the data to be operated upon.

Instructions given to a human computer must be fairly complete. Some instructions may be implicit; that is, they have been supplied to him previously. For example, he might know a procedure for extracting a square root. He would not, however, be expected to invent such a procedure. A machine computer needs even more complete directions. It must be instructed in every detail of the intended operation. This instruction is, in part, supplied in the construction of the machine and, in part, in the instructional input for the particular job. It can never be told, "Use your own judgment." To make the discussion uniform, we may similarly restrict the definition of a human computer; when he responds to such a vague directive he is not functioning as a computer but in some other capacity.

The description can now be sharpened as follows: A computation is the transformation of a body of information. Its output is completely determined by its direct input together with the well-defined rules of operation of the computer. A computer can transform but cannot produce information. In the recently developed theory of communication, there appears a close analogy with thermodynamics. Information plays a role like "negative" entropy. In a computation, information can be, and usually is, lost; no information can be created. This seems to cast doubt on the usefulness of computers. If each transaction may show a loss but never a gain, where is the profit? However, the *amount* of information as defined in communication theory is not the only measure of its utility. Some computations present information in a more useful form by expansion. Thus, a harried navigator finds more comfort in a table of haversines than in a succinct formula which implies all the numbers in that table and infinitely many

---

A special article recommended for publication by the AIEE Committee on Computing Devices.

Stanley Frankel is a consultant to Librascope, Inc., Glendale, Calif.

more besides. In other uses, a computer refines a large mass of data into a smaller but more useful summary. The computer may then be likened to a smelter yielding a compact body of virgin metal which commands a higher price than the gross ore.

The ability of a computer to transform instructions and primary data into a desired set of consequences has been well described as follows: "The Analytical Engine has no pretensions whatever to *originate* anything. It can do anything we *know how to order it* to perform. It can *follow* analysis; but it has no power of anticipating any analytical relations or truths."[2]

The output of a computation can be regarded as the solution to a problem posed by the input. The problem may be a trivial one, such as a test problem which is run to see if the computer is behaving properly, but still it is a problem to the computer, if not to the operator. Roughly speaking, the value of a computer is indicated by the extent of the set of problems it can solve. Here we must distinguish a solution from a mere answer: whatever the input, the computer will "answer" by doing *something*, but that answer may not be a solution to the problem the operator has in mind. Not every problem can be solved by a computer; in fact, some problems have no solutions. More precisely, we should say that there is no way of putting these problems to a computer which will produce as output a satisfactory solution. We are not concerned here with the difficulty in finding a way of posing a problem. That is a matter of the ingenuity of the user. We ask only whether or not the required solution can result from *any* input to the computer. In particular, we are led to ask about the class of problems which can be solved by a "best possible" computer.

This question was discussed by A. M. Turing in 1936.[3] He describes a computer as a "black box" of finite complexity; that is, one capable of existing in a finite set of states. It operates on a finite body of information, its input, initially recorded on its memory unit which has indefinite capacity. It will be seen later that this is a very practical idealization of existing digital computers. The output of the computer, the answer to its problem, is an indefinite series of digits reported out in the course of its operation. By regarding these digits as the expansion of a real number, we may think of the answer to any computational problem as a single real number. A number which could be produced in this way by any possible computer is called a "computable number." The set of all solvable problems is thus represented by the set of all computable numbers. It is a consequence of the required finiteness of the complexity of a computer and of its input that the computable numbers form a merely denumerable set. (Thus most numbers, or in the language of measure theory "almost all" numbers, are noncomputable. It is, nevertheless, quite difficult to produce an example.)

One remarkable result of Turing's investigation is that he was able to describe a single computer which is able to compute *any* computable number. He called this machine a *universal computer*. It is thus the "best possible" computer mentioned.

The machine described by Turing is able to hold in its memory a suitably coded description of an arbitrary computer together with its input, and is able to simulate the behavior of that computer. For any solvable problem a particular computer could be described which solves it. By simulating that particular computer, the universal computer can also solve that problem. Thus, the universal computer is able to solve any solvable problem. This surprising result shows that in examining the question of what problems are, in principle, solvable by computing machines, we do not need to consider an infinite series of computers of greater and greater complexity but may think only of a single machine.

Even more surprising than the theoretical possibility of such a "best possible" computer is the fact that it need not be very complex. The description given by Turing of a universal computer is not unique. Many computers, some of quite modest complexity, satisfy the requirements for a universal computer. In particular, it will be seen in the following that any of the modern general purpose computers, such as the relatively simple *LGP-30*, is a universal computer, as is the "Analytical Engine" mentioned previously.

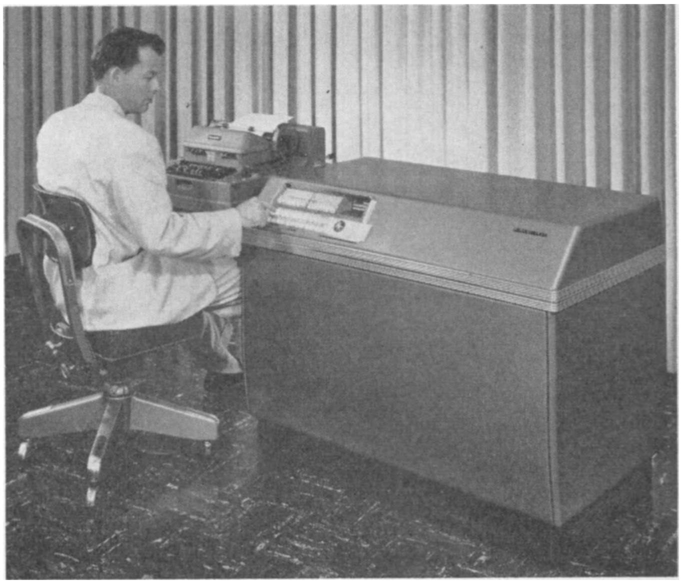

Fig. 1. The LGP-30 computer, including a Flexowriter as input-output device.

We now have a partial answer to our question as to the range of problems which can, in principle, be solved by a general purpose computer (GPC); namely: What one GPC can do so can another. This statement has to do only with what problems a GPC can, in principle, solve; leaving aside such practical considerations as speed, cost, and reliability. Naturally, the utility of a GPC for a particular application depends on its ability to produce reliable results quickly and cheaply as compared with competing computers both human and machine. In the following sections, the structure and mode of operation of the *LGP-30*, as an example of drum computers, will be described in some detail. It can then be seen in what ways these practical considerations restrict the range of problems to which the drum computers can usefully be applied.

THE FUNCTIONAL DESIGN

THE CHIEF MEMORY DEVICE of the *LGP-30*, like other drum computers, is a "magnetic drum." This is a rotating cylinder having a magnetizable surface on which information can be recorded in much the same way as music is stored by a magnetic tape recorder. The body of information stored in the memory of a GPC is usually held as a set of segments, each of which can be "read" and brought into active participation in the computation as required. The instructional information usually consists of a series of elementary instructions which are read and obeyed one after another. For example, a typical instruction in the *LGP-30* calls for the performance of an arithmetic operation on two numbers. One of these numbers is selected out of the body of numerical information in storage by designating the memory "location" in which it is held. The other number used in the operation is that which is held in a specialized memory device called the accumulator. At the end of the operation, the result is retained in the accumulator. The accumulator consists, in part, of two heads close to the rotating drum in the same circumferential track, together with associated electronic equipment. These two heads and a third, which is discussed later, are shown in Fig. 2. The first head records on the magnetizable coating of the drum a sequence of (binary) digits. Each digit is, at a slightly later time, read by the second head and presented to the active circuitry of the computer. During some phases of activity, digits are recorded by the first head exactly as they are presented by the second head. This leads to the recirculation of the sequence of digits held at any one moment on the magnetic coating of the drum between the two heads. For this reason the accumulator is called a circulating register.

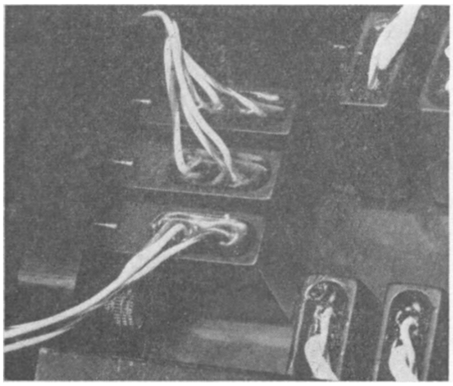

Fig. 2. The recording and two reading heads of the accumulator.

To prevent progressive deterioration of the quality of recording, it is necessary to sharpen and "retime" the recorded digits with each recirculation. Another track of the drum is used to provide the basis for this retiming. On this sprocket track there is a permanently recorded periodic signal which is read by a head, amplified, peaked, and presented to the active circuitry as a regular sequence of *clock pulses*. The interval between two consecutive clock pulses is called a *digit period*. A digit period occupies about 8 microseconds. In each drum revolution (occupying about 16 milliseconds), there are 2,048 digit periods. At each clock pulse, the amplified output of the accumulator reading head is sampled and used to set a *toggle* (i.e., a flip-flop, a bistable multivibrator) either to its "zero" or to its "one" state. This toggle holds and presents the binary digit so read during the succeeding digit period. This binary digit (bit) is recorded by the accumulator recording head during that digit period if simple recirculation is required. The distance between the recording and reading heads of the accumulator is such that the period of recirculation is 32 digit periods; in other words, 1/64 of a drum revolution.

Three other permanently recorded timing tracks produce various pieces of timing information. One of these is a signal which distinguishes every 32-digit period, called a *sign digit period*. A group of 32 consecutive digit periods ending with a sign digit period is described as a *word period*. Similarly, the 32 digits presented by any part of the computer during a word period are called a *word*.

The word presented by the accumulator usually represents an algebraic number by the following system of interpretation: The first digit is a *spacer bit* of no numerical significance. (It is always recorded as a zero.) The second digit presented has the numerical value 2^{-30} if the digit is 1, otherwise 0. The third digit presented contributes the numerical value 2^{-29} or 0. This progression continues up to the 31st digit which contributes the numerical value 1/2 or 0. However, the final digit, presented during the sign digit period, does not have the numerical value 1 or 0, but rather the value -1 or 0. This permits representing both positive and negative numbers.

The main memory of the *LGP-30*, usually called simply the *memory*, is held on 64 tracks of the drum, each with an associated head. Any one of these heads can record information on its track or read from it information previously recorded, though not both simultaneously. Only one reading circuit and one recording circuit is provided for the memory. Which one of the 64 heads is connected with these circuits is determined by the settings of six P-toggles, In reading from the memory, the selected head sets a toggle, V, in each digit period. Toggle V presents, in succession, the 2,048 bits previously recorded in that track. These bits form the 64 words of 32 bits each held in that track.

The particular word of the memory to be used in an operation is singled out in two steps: First, by setting the P-toggles, the required track is selected. Then, after waiting an appropriate number of word periods, the desired one of the 64 words in the selected track is ready to come under the head in the word period in which the operation is performed. The desired word (or, more precisely, the memory location in which it resides) is designated by its *address*. An address consists of two (6-bit) numbers, each having the range 0 to 63. One, the *track number*, indicates the track to be selected. The other, the *sector number*, selects the required word of that track. In most operations, some one memory location is selected and sought out in this way. In a *record* operation a word, or part of a word, is recorded on the selected track during the word period of execution. Each instruction includes an address, its *operand address*, designating the memory location to be selected.

The *LGP-30* has two other circulating registers, besides the accumulator, each having a recirculation time of one word period. One of these is the *counter*. Its primary purpose is to hold the address of the memory location in which will be found the next instruction to be obeyed. After an operation has been completed the memory location designated by this *control address* is sought out in the same way as an operand memory location is sought. The word found at the control address is read into the third circulating register, the *instruction register*, where it is held for as long as is required for it to be acted upon as an instruction.

An instruction word has two parts, an order and an address. Four bits describe the order, indicating the kind of operation to be performed. Six bits indicate the sector number and six more the track number of the operand address. By reason of this structure of an instruction the *LGP-30* is said to have a *one-address code*.

After an instruction word has been located and set into the instruction register, the control address, held in the counter, is augmented by one to prepare for its next use. (In the normal course of events, instructions are read and executed from consecutively addressed memory locations.) Thereafter the content of the counter usually recirculates without change until after the next reading of an instruction word. However, if an instruction is an effective *control transfer instruction* its execution consists of the transfer of the content of the instruction register into the counter. This produces a deviation from the normal sequence of instruction locations described in the foregoing.

After an instruction has been set into the instruction register, it is held there until several activities are completed. During the next word period, the order part of the instruction is set into four Q-toggles and the track number of the operand address is set into the P-toggles to prepare for the reception of the operand word. The operand sector number usually cannot be disposed of summarily. In succeeding word periods, this sector number, recirculating in the instruction register, is compared with a synchronously presented announced sector number, provided by the timing tracks. Each of the 64 sector numbers is announced in one of the 64 word periods of a drum revolution. The sector numbers are not announced in numerical sequence but in accordance with an *interlace pattern* described later. Agreement, between the announced sector number and that recirculating in the instruction register, marks the end of the period of search. In the following word period, the operation is performed or, if the operation is a multiplication or division, commenced. After this agreement has been found, the instruction need no longer be held in the instruction register. Accordingly, this register is used, on multiplication or division, to hold for later use the operand word (the multiplicand or divisor) which is set into it in the following word period.

THE ELEMENTARY OPERATIONS

THE 4-BIT ORDER CODE used in the *LGP-30* permits the use of 16 elementary operations. All of the 16 order codes permitted by the code structure are used. Some of these orders may be explained as follows:

The order, *extract*, is used to isolate selected portions of a word, replacing all digits of the remaining part by zero. It can be used in packing several discreet brief pieces of information into one word and later separating them for individual use. This may be desirable for economy of information storage or because the separate pieces of information are closely related in a way which makes it convenient to handle them together, e.g., a month, day, and year which together specify a date. Another use for *extract* arises in the use of a function table stored in the memory. If an argument for this function is determined, in the course of the calculation, it may be used to get a value from the table by arithmetic manipulation and abridgment by the use of *extract*, thus forming an address to be set, say, into a *bring* order. Similarly, the remainder can be separated for use in an interpolation. An address which has been produced by arithmetic processes and by use of the *extract* order can conveniently be set into an instruction by use of the *y* order.

In multiplying, the accumulator is extended to double length by the use of an additional reading head which is spaced 33 digit positions beyond the first reading head. This extension gives space for a complete product consisting of a sign and 60 digits. After completion of the multiplication, the normal accumulator recirculation in one word period is resumed; hence, only part of this complete product can be retained. For the order *M*, the more significant half is retained, the sign and 30 bits. For the order *N*, the less significant half of the product digits are retained.

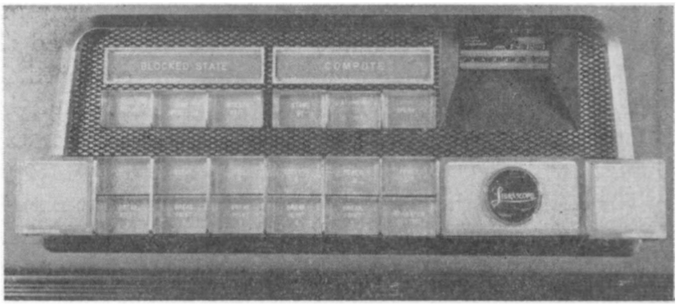

Fig. 3. The control console of the LGP-30 and the oscilloscope display of the contents of the circulating registers.

Information is set into the *LGP-30* with the help of a Flexowriter, a punched paper tape controlled typewriter. A previously prepared tape may be used to set information into the accumulator, from which it is later transferred to an appropriate memory location under the control of the computer. In the *input* process, four digits of a tape symbol are inserted into the accumulator the previous content of the accumulator being moved "upward" to make room.

The conditional control transfer order, *test*, is used to introduce a "branching" of the sequence of instructions, contingent on the results of prior computation. It may, for example, be used in computing a function by successive refinements. After each refinement is completed, some measure of error is computed and subtracted from a preassigned tolerance value. A test order will then return control for another cycle of refinement if this difference is negative, showing that the error is intolerable, otherwise go on to further operations.

The conditional stop order, *Z*, helps in finding the errors in a malfunctioning "program." It may stop the computation, depending on the settings of several switches and of the P-toggles, set by the track number part of the address. The address has no other significance. By scattering *Z*-orders with various track numbers in the program (the sequence of instructions), the operator is enabled, by setting switches, to stop the computation at a point of his selection. Then by use of the one-operation mode he is able to go through the following part of the computation at a slow rate. In the one-operation mode, the computer, when started by a push-button, reads and obeys one instruction and then stops. The contents of the three circulating registers are constantly displayed on an oscilloscope to facilitate checking. This oscilloscope display of the registers, together with the operator's "console" of the *LGP-30*, is shown in Fig. 3.

Output also used the Flexowriter, which is then controlled by the computer. The order *print* causes the Flexowriter to type one symbol (or to space, carriage return, shift case, etc.) and, if desired, to punch that symbol on a paper tape. The effect of a *P*-instruction is determined by the track number part of its "address." There are, thus, 64 different print orders which may, by suitable programming, be used to produce any record which the Flexowriter is equipped to type. They may also be used to record results on tape in a form which permits it to be reinserted in the computer at a later time.

The ability of the *LGP-30* to punch intermediate results on a tape which can later be reinserted for further use gives it, in effect, an infinite memory capacity. The use of this auxiliary storage medium seems not entirely automatic since an operator is required to handle the tape. We may, however, think of the operator as acting under instructions supplied by the computer. (It is, for example, perfectly feasible to include in the program provision for typing out instructions to the operator in clear English.) It is in this sense that the *LGP-30*, which like any other general purpose computer has only a finite *internal* memory capacity, may be regarded as a universal computer.

SPEEDS OF OPERATION

In several drum computers, including the *LGP-30*, a drum revolution occupies about 17 milliseconds. Since executing an order requires first awaiting the appearance of the instruction, and then of the operand word, a typical time per operation would be in the tens of milliseconds. Several devices are used in drum computers to reduce this mean operation time. In some, such as the International Business Machines Corporation (IBM) *650*, a considerably higher drum speed is used. In some, like the Datatron, a part of the memory is held in recirculating loops which have shorter mean "access times." In the *LGP-30* another device is used, the aforementioned interlace pattern. Memory locations are so spaced around a track of the drum that eight word periods elapse between the presentations of two consecutively numbered words (in particular, two consecutively obeyed instructions). Problems may be so planned that the operand word usually appears in one of the middle six of these word periods. Planning a problem in this way is called "minimum latency coding." By exercising moderate care in coding, the mean time per operation in a typical problem may be brought down to about 4 milliseconds for the brief operations and 20 milliseconds for multiplications and divisions.

PROBLEMS SUITABLE FOR A DRUM COMPUTER

It has been argued in the foregoing that the class of problems which a drum computer such as the *LGP-30* can solve *in principle* is just the same class as for any other GPC, such as Charles Babbage's Analytical Engine referred to in the previously quoted statement. The statement, written in 1842 by Ada Augusta, the Countess of Lovelace, remains today an excellent description of the problems which can be solved by a Univac* *701*, or *LGP-30*. There remains the practical question, "What problems would it be *sensible* to present to a drum computer?"

As a partial answer to this question we can exclude two types of problems. At one extreme, there are computational problems which are too simple to justify the use of any automatic computer. If a calculation can be completed by hand in a few hours it is not likely to be worthwhile to devise a procedure for performing it on an automatic computer. Here we assume that this is a "one shot" calculation, not one which will be repeated many times with only minor variation. At the other extreme lie problems of such great computational difficulty that they cannot be completed in reasonable time by a drum computer. (It is assumed that such a problem is "all of a piece," i.e., it cannot be separated into parts which could be parceled out to many drum computers.) A classic example is the computation of tomorrow's weather on the basis of today's observations. The prediction will be useful only if it can be produced in less than a day. Some of these problems can be treated usefully by large-scale electronic computers which may have computing speeds several hundred times that of the drum computers. Others cannot reasonably be handled by any existing computers.

Between these extremes lies a broad class of problems which might sensibly be presented either to a drum computer or to a large-scale computer. (The two classes of computers have comparable speed-to-cost ratios.) In this range of problems several circumstances may favor the use of drum computers. One arises from the requirements of maintenance. The large-scale computers, by reason of their much greater complexity, may be expected to be subject to more frequent breakdown. It is also to be expected that a greater time will be required to track down and replace a malfunctioning component. This consideration bears most strongly on operations in which continuity of service is important. A single giant computer is more vulnerable to tube failure than an equivalent host of midget types.

For problems having considerable logical complexity but not requiring particularly great computing time, the drum computers have an advantage in the possibility of using the computer in checking out the program. In working through parts of a problem to find the origin of a difficulty,

---

* Reg. U. S. Pat. Off.

the greater speed of a large-scale computer is of no advantage. It usually is not economically feasible to use the large-scale computers in this way. The programmer can reasonably use a drum computer to assist in his part of the work, whereas an inexpensive man tends to become the slave of an expensive machine.

Drum computers have an evident advantage in small shop operations, where the work load is not sufficient to justify the presence of a large-scale computer. This is particularly true in places remote from service bureaus where the part-time use of a computer can be obtained. The same consideration applies to large but dispersed operations in which computing can be done more conveniently in many locations than in a central computing installation.

The discussion in the first part of this article showed that the class of "computations" is not one which can be narrowly described. A great variety of operations, arithmetic and otherwise, can be performed by computers. (As an extreme example, Turing examined the feasibility of sorting all possible mathematical assertions as true or false. This task, however, proved to be beyond the ability of even a universal computer.) In this section, an attempt has been made to indicate which problems are suitable for a drum computer by excluding broadly described classes. Some problems are too simple, some are too long for drum computers. In the following, a few examples of problems which are suitable for drum computers are displayed. They do not, of course, cover all of the uses now being made of drum computers, nor those uses yet to be found.

The preparation of a mathematical table typifies many problems in which a drum computer can be used to good advantage. Each step of the computation may be quite simple; for example, each entry in a table of exponentials can be obtained from the preceding entry by one multiplication. Almost all of the mathematical tables which have been published could now be produced quite expeditiously by drum computers—in most cases as fast as the results can be typed. For example, the excellent tables prepared by the Federal Works Agency of the Works Progress Administration in New York City could now be duplicated by an *LGP-30* at a rate of about one volume per day. Many business and accounting problems are of this type. The preparation of a large payroll by hand is a burdensome task even though each individual computation is of quite moderate complexity.

Some information-handling activities are difficult to carry out by hand by reason of the need for quick access to a great body of stored information even though no burdensome amount of arithmetic work is involved. Drum computers are beginning to be used for airline reservation reporting and other inventory reference problems. Probably many drum computers will be so used in the future. These machines tend to be specialized in having large internal memory capacity and many channels for input and output but have less arithmetic facility than other GPC computers.

Many tasks now being performed by computers previously went undone because of the great difficulty of doing them by hand. The recent development of "linear programming" and other methods of operations research is, in part, owing to the availability of computing machines. These techniques permit making many managerial decisions on the basis of calculation where formerly they could be based only on intuition.

Many problems involve the study of fields, electric, magnetic, acoustic, etc., which are governed by simple differential equations but are not analytically tractable. Some such problems which formerly could be treated only approximately, by field plotting methods and the like, can now be dealt with accurately and conveniently by drum computers. Others, of greater complexity, require the faster large-scale computers or are beyond the reach of any existing machines.

The techniques for preparing problems for solution by a digital computer, "programming" and "coding," are by now fairly widely known. Fortunately, the basic operations and abilities of the various GPC computers are very much alike, hence a person who has become familiar with the management of one computer can easily shift to another. For many computers, like the *LGP-30*, the elementary operations are suitable for a fairly straightforward translation from mathematical representation into the language of the computer. For example, the evaluation of $(ax+b)/c-d$ is fairly readily represented in the *LGP-30* code as *Ba*, *Mx*, *Ab*, *Dc*, *Sd*, where *a*, *x*, etc., are to be replaced by their addresses. The more sophisticated techniques required for efficient coding for each computer are fairly easily learned. Engineers only casually involved with the use of a computer can quickly pick up these skills, such as the use of minimum latency coding, or may disregard them for problems which do not make severe demands of computer time. Thus, the use of a drum computer should not be importantly limited by difficulty in finding or training its users.

REFERENCES

1. LGP-30 General Purpose Computer, **S. Frankel, J. Cass.** *Instruments and Automation*, Pittsburgh, Pa., vol. 29, 1956, p. 264.

2. Faster than Thought (book), edited by **B. V. Bowden.** Pitman & Sons, Ltd., London, England, 1953, p. 398.

3. **A. M. Turing.** *Proceedings*, London Mathematical Society, series 2, vol. 42, 1936, p. 230.

New Cancer Diagnosis Technique

Instruments being used in the development of the guidance equipment for the United States Air Force *SM-62* Snark missile are now aiding in the search for improved cancer diagnosis. New techniques in microscopy, more advanced than medical research has found, have been developed by Northrop Aircraft, Inc., Hawthorne, Calif. Particles as small as 10^{-12} gram can be chemically analyzed with adaptations in these techniques.

Using specially designed lenses and electronic circuitry, medical researchers are able to detect the smallest light variations produced in a specimen by the presence of infinitely minute particles of matter. The lenses are capable of being used even in the invisible sections of the spectrum.