NEOVIUS, G. (1955). "Artificial Traffic Trials using Digital Computers," *Ericsson Technics*, Vol. 2, p. 279.
THE RAND CORPORATION (1955). "One Million Random Digits and 100,000 Normal Deviates," The Free Press, Glencoe, Illinois.
TAUSSKY, O., and TODD, J. (1956). "Generation and Testing of Pseudo-random Numbers," Symposium on Monte Carlo Methods, University of Florida, 1954, Wiley, p. 15.
THOMSON, W. E. (1958). "A Modified Congruence Method of Generating Pseudo-random Numbers," *The Computer Journal*, Vol. 1, p. 83.
THOMSON, W. E. (1959). "ERNIE—A Mathematical and Statistical Analysis," *J. Roy. Stat. Soc., Ser. A*, Vol. 122, p. 301.
TIPPETT, L. H. C. (1927). *Random Sampling Numbers*, Cambridge U.P.

Time-Sharing on the National-Elliott 802

by R. L. Cook

Summary: This article discusses some of the programming techniques that have been evolved at Elliott Brothers for dealing with on-line applications of the 802 computer in the process-control field. Time-sharing or program interruption methods are described with particular reference to a straightforward data-logging application.

THE 802 COMPUTER

The 802 was designed for two purposes: to perform as a small general-purpose computer, and also to act as the computing centre for on-line process-control systems.

The logical elements of the 802 consist of a junction transistor-magnetic core element shown in Fig. 1. By this means, the high reliability and the little or no maintenance that are required in on-line applications can be achieved.

The 802 has a magnetic-core store capable of holding 1,024 words, each of 33 binary digits. The size of the store can be extended to 4,096 words if necessary. The order code comprises 64 basic orders, most of which refer to the single accumulator and one specified store address. Great care has been taken to ensure that the order code is simple to learn, consistent, and without exceptions. Each 33-bit word can hold two 16-bit orders, together with a B digit. If the B digit is present, the first order is obeyed in the normal manner, but the (new) contents of the store address specified by this order are added to the second order before it is obeyed.

In this way any location of the store may be used as a B-modifier. If the B digit is absent, the two orders are obeyed sequentially without B modification.

There are three independent input channels on the 802: channel 1 is normally attached to a paper-tape input device; channel 2 can receive information, via a switch, from any number of different input devices; and the third channel consists of a set of manually-operated keys, called the *Number Generator*, that enables a single word to be entered. The switch on channel 2 is operated by program control, and by this means, any number of different devices, each capable of giving digital information, may be simultaneously attached to the 802 and individually switched. Two output channels are provided from the computer: channel 1 is fixed and normally

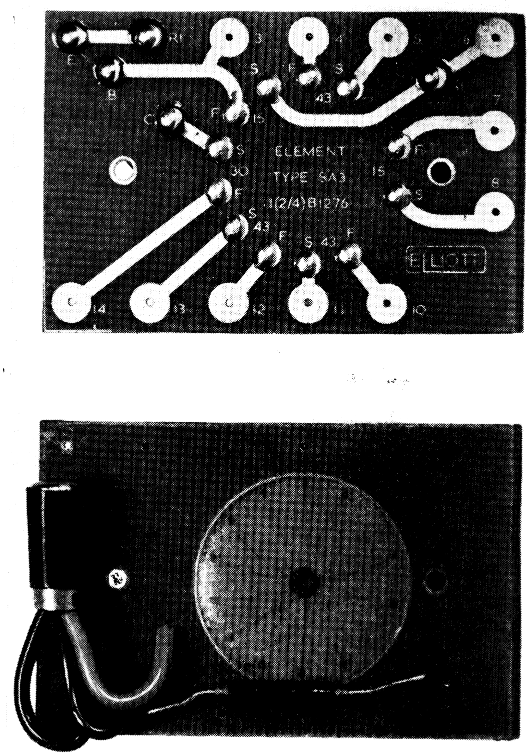

FIG. 1.—The junction transistor and magnetic core which make up the logical element of the 802. The use of solid state devices ensures the high reliability required for on-line working, in addition to minimizing the space and power required for the computer.

operates a paper-tape punch, and channel 2 can be attached by means of a switch to any device capable of accepting digital information.

The completely transistorized 803 computer, which is functionally identical to the 802 except that it has a 4,096-word core store and a 39-digit word, is also used for process-control applications in the manner described in this article. It forms the computing centre of the Panellit 609 Process Control System.

CONTROL APPLICATIONS

In a control application, the 802 works on-line in real time. By *on-line* we understand that information specifying the state of the system is fed directly into the computer. This information may consist of measurements of voltage, temperature, pressure, flow, etc., together with various control signals which indicate the action that the 802 has to take.

The output from the computer may be required in printed form, as a visual display, or, in some cases, to control the system directly and so close the loop.

The signals from measuring instruments are mostly analogue in form and must be digitized before being fed to the 802. A special analogue-to-digital converter has been developed for use with Panellit 609 process control applications. This will accept an analogue input voltage in the range 0–10 volts; the output is the 12-bit digital representation of the voltage, with a discrimination of one part in 4,096. The signals must be brought to a common voltage range before entering the converter. In some cases, when the analogue signal is not proportional to the value to be measured, it is convenient to linearize the signal whilst it is in analogue form, rather than to perform this process digitally either by program or hardware.

In most control applications the 802 works on a fixed program, the operation of which is entirely controlled by the input signals received from the measuring devices—although the manual-control keys will be used on occasion to request particular action by the computer. A digital clock may be attached as an input device to enable operations to be undertaken at fixed times. The clock can be made to send signals to the computer at fixed time intervals so that if, for example, a particular input channel must be read every 10 seconds, the clock will be made to send a signal, lasting perhaps $\frac{1}{4}$ second, every 10 seconds.

A TYPICAL PROCESS APPLICATION

The programming aspects of a process application will be discussed with reference to a typical data-logging system. The computer and its equipment are shown in Fig. 2. The computer must carry out the following operations.

(*a*) Scan the input points at fixed intervals of time, integrate and perform other calculations on this data, and test for alarm conditions if the value of a point falls outside a given range.

(*b*) Log the values, i.e. print out summaries of the values of the points each hour.

(*c*) Perform special operations as determined by the Manual Control. These will include printing special summaries, and altering the permitted range of certain points.

(*d*) Other work. Whilst the data connected with the process is held in the computer, and provided that the computer is not fully occupied on the control aspects of the project, it should be used to perform other calculations connected with the running of the plant. These might include calculations of total throughput, piece-work bonus calculations, calculation of efficiencies or other plant statistics, experiments on the effect of other methods of control, and determination of load criteria, e.g. the determination of which process should be shut down when the output exceeds the demands.

In many data-logging applications, it is the base load (*d*) which justifies the use of a computer rather than conventional data-logging equipment. A computer is an expensive and complex mechanism that is capable of accurate and lengthy calculations, and these attributes must be exploited to the full. It can work faster and more accurately, but not more intelligently, than a human being, and it can provide faster and more accurate control than by human means.

TIME-SHARING

There are two ways of ensuring that a computer will be ready at a certain moment in time to perform a calculation; one is to build a computer that is sufficiently fast to enable it to have completed the previous calculation by this time, the other is to arrange for the previous calculation, if not complete, to be interrupted and completed at a later time. The former process is uneconomic since it implies the construction of a computer that is faster, and hence more expensive, than is really required. The latter process is the principle of *time-sharing*.

Time-sharing, or *program interruption*, on the 802 is performed by program control. The 802 has, in effect, to perform, apparently simultaneously, several programs. However, except for the base load, none of the programs have to be performed unless a demand is made. This demand may be a time signal from the clock, the depression of a manual key, or the result of a calculation performed in another program—for example, an alarm may be demanded as a result of an input being out of a prescribed range. Each demand consists of a digit or group of digits, and these are staticized and fed into the 802 as a single word of information, called the *control word*, via the number generator input channel. The number generator itself is not used during the operation of the program, but since it is in parallel with the control word, it may be used to represent the demands manually whilst the program is being tested.

The presence of any particular digit in the control word is interpreted as a demand for the operation of the associated program. The programs are allocated an order of priority, and the 802 is made to obey at all times the program of highest priority that has been demanded. At periodic intervals the control word is examined for the presence of a further demand. The

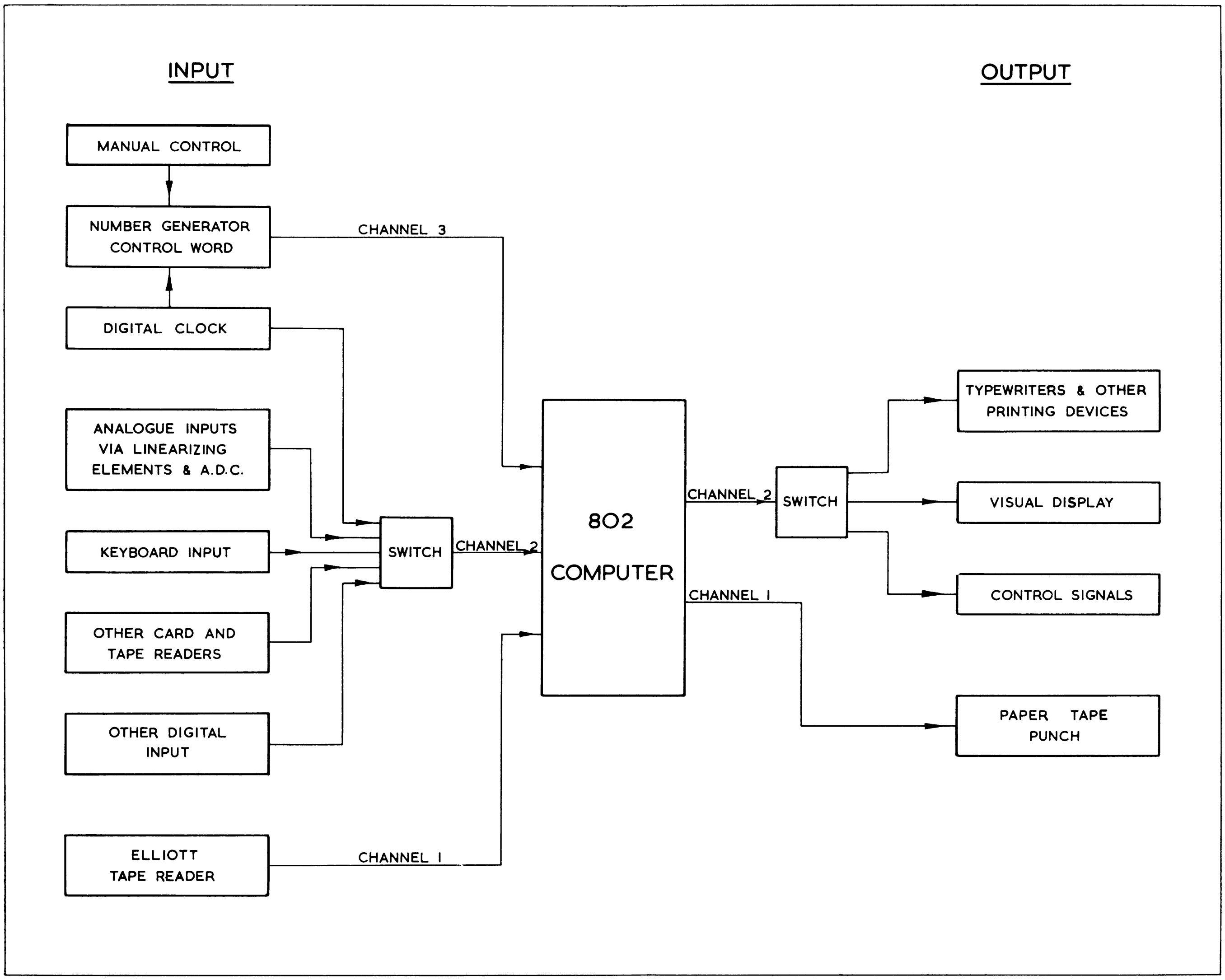

FIG. 2.—Equipment used in a typical process application. Input and output on the channels 2 can be independently switched by program control.

control word may also be used to test the state of electro-mechanical devices by feeding their *busy* lines (or the inverse) to digits in the control word. In general it is the presence of a digit in the control word that indicates that action is to be taken by the computer. In the case of electro-mechanical devices it is the inverse of the busy line that is fed to the control word, so that the presence of a digit indicates that the device is free for use. In this way it is possible to ensure that, for example, the typewriter is not busy before giving a print order.

It happens that most of the different programs are short and can be completed in a time comparable with the speed of operation of the typewriter or other electro-mechanical devices. In this case, the control word is scanned at intervals during the operation of the long programs, but only on the completion of the shorter programs.

EXAMPLE

The data-logging application mentioned above will now be described in detail. The basic programs associated with each control digit are, in order of priority:

(1) Store manual demand keys and set 6.
(2) Read the previously selected input channel and switch to the next input channel: set 4.
The switching between the input channels is performed by means of a set of mercury wetted-contact Clare relays.
(3*a*) Print one character on first output device.
(3*b*) Print one character on second output device.
(4) Process last point read and set 5 if necessary.
(5*a*) Calculate next character for first output device.
(5*b*) Calculate next character for second output device.

(6) Take action on manual demand and set 5 if necessary.
(7) Read time (for hourly summary).
(8) Base load: long-term calculation of efficiency, plant statistics, etc.

Items (3) and (5) are repeated for each output device that is currently in use. In addition to these operations, the control word must be examined about every 100 msec and, if necessary, the priorities must then be reallocated. Care must be taken that each digit that appears in the control word is read once and only once. The control word is also examined on completion of each operation.

The programmer has control over the possible points at which a program interruption can occur, and these can usually be chosen so that the content of the accumulator is no longer required. The content of the sequence control register is, therefore, the only information that must be stored before a program interruption. At any moment of time, the complete state of the calculation is specified by a priority list giving the addresses at which each operation must be recommenced, together with a set of digits showing those operations which have been demanded but have not yet been completed. This set of digits can be stored as a set of sign digits in the priority list. Whenever an operation is completed, the sign digit is removed and the address put equal to the starting address of the operation. The master controlling subroutine then simply consists of a program to test the control word and to insert a sign digit into the words in the priority list corresponding to each digit present. Control is then transferred according to the word in the priority list which corresponds to the highest priority and which is also negative. The programmer can write and test the program for each operation separately. The time-sharing operation then consists of inserting entry orders into the master controlling subroutine, at periodic intervals throughout each of the programs. We are at present considering writing a simplified programming procedure that will insert these interruption points in a proved program, and thus simplify even more the writing of time-sharing programs on the 802.

Great care must be taken in choosing the order of priorities. In the example, delay in obeying operations 4 to 8 has no serious effect, although item 4 must be obeyed between each reading of the scanned inputs. It has been found that in some applications it is necessary to alter the priority according to the conditions—for example, if a manual command has not been obeyed within a short while of the demand, it is automatically given a higher priority. In many applications 50% of the total time is available to obey the base load.

CONCLUSION

Time-sharing has been found necessary in all process-control applications which have been considered. It provides the following advantages:

(1) It allows the computer to deal with inputs that can occur at random, undetermined, moments of time.
(2) The most efficient use can be made of the computer by the use of the priority system. At moments of crisis only essential work will be done, the less important work being performed later on, when time is available.
(3) Machine time can be saved by allowing the inverse of busy lines to act as demand digits and thus enable electro-mechanical input and output devices to be interleaved.

Considerable thought has been given to the possibility of making program interruption automatic by means of extra hardware. An automatic system would reduce the length of a program slightly, but, at the moment, we feel that it would be impossible to design such a system that would deal economically with the complicated priority conditions that occasionally arise.

ACKNOWLEDGEMENT

The author wishes to express his thanks to Elliott Brothers (London) Ltd. for permission to publish this paper.